Commodore
COMPUTER CLUB
38
L. 4.000
La rivista degli utenti di sistemi Commodore
25 gennaio 1987 - Anno VI - N. 38 - Sped. Abb. Post. GR. III/70 - CR -Distr.: MePe
Incontri ravvicinati col GW-BASIC '64
Spazio "AMIGA"
La fabbrica delle adventure
Metti un tigre nel 128
Koala per C16
Totocalcio a tutto 64
INSERTO: La funzione Random
systems

SUL SENTIERO DELLE GIUBBE ROSSE

Una vera esperienza di vita per i ragazzi/e oltre i 10 anni
Abbinare lo studio della Lingua Inglese, al contatto di una natura incontaminata
Una vacanza-studio unica ed indimenticabile, in uno scenario che non ha confronti.

Questo tipo di vacanza è indirizzato sia ai principianti, sia a coloro che hanno già maturato una conoscenza della Lingua Inglese, ma il denominatore comune è il reale contatto con la natura.

- Un viaggio di 19 giorni attraverso la Provincia dell'Ontario a bordo di un "Super Van" da 15 posti, con aria condizionata e stereo system, con l'assistenza di personale qualificato.

Ogni 10 partecipanti ci sono 4 persone di assistenza.
Le attività standard includono:

- partecipazione alla vita di campeggio, canoa, tracking, pesca, white-water rafting, ginnastica, nuoto e studio della Lingua Inglese.

Quest'ultimo aspetto sarà curato da insegnanti specializzati dello SHENKER INSTITUTE OF ENGLISH, con due ore di lezione al giorno, seguendo il METODO SANDWICH di GEORGE SHENKER.

- Viaggio Andata/Ritorno MILANO/TORONTO in classe turistica.
- Tre pasti al giorno dalla 1ª colazione del secondo giorno, al lunch del 17 giorno.*
- Full-Day Immersion di Lingua Inglese con personale SHENKER più 2 ore al giorno di corso intensivo.
- Materiale audio-didattico per il Corso comprendente:

* Walkmann
* Cassette
* Libri di testo e work book

- Assicurazione completa EUROPE-ASSISTANCE.
- Pernottamenti in hotel di categoria superiore e in Campeggi nei "NATIONAL PARKS".
- Tutte le tasse d'iscrizione.

attrezzatura da pesca
* canna da pesca con mulinello
* licenza di pesca
I costi di noleggio del Super Van e tutti i costi annessi:
* benzina
* autostrade con pedaggio

Il programma si divide in 4 diversi turni a partire dall'ultima settimana di giugno
Prenotazioni e informazioni presso:

SHENKER INSTITUTE OF ENGLISH - Corso Monforte, 36 (MI) - Tel. 02/700332/700363/700929
ore ufficio - Sig.ra Sawchik - Olivieri (ore serali) Tel. 039/513211

UVET - Viale Ferdinando di Savoia, 4 (MI) - Tel. 02/675061 (30 linee)
ore ufficio - Sig. Biagi

SYSTEMS - Viale Famagosta, 75 (MI) - Tel. 02/8467348/9
ore ufficio - Sig. Tidone

in collaborazione con:
SHENKER INST. OF ENGLISH
CP AIR
UVET

Sommario

INSERTO

LA FUNZIONE RANDOM: APPLICAZIONI PARTICHE

RUBRICHE

Direttore: Alessandro de Simone
Redazione/collaboratori: Claudio Baiocchi, Carlo e Lorenzo Barazzetta, Giovanni Bellù, Simone Bettola, Andrea e Alberto Boriani, Diego e Federico Canetta, Giancarlo Castagna, Umberto Colapicchioni, Pasquale D'Andreti, Maurizio Dell'Abate, Valerio Ferri, Luca Galluzzi, Michele Maggi, Giancarlo Mariani, Marco Miotti, Flavio Molinari, Claudio Mueller, Massimo Pollutri, Carla Rampi, Fabio Sorgato, Giovanni Verrelli, Antonio Visconti.
Segreteria di redazione: Maura Ceccaroli.
Ufficio Grafico: Arturo Ciaglia
Direzione, redazione, pubblicità: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Giorgio Ruffoni, Claudio Tidone - V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
- Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
- Toscana, Marche, Umbria: Mercurio Srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
- Lazio, Campania: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8109679

Segretaria: Marina Vantini
Tariffe: prezzo per copia L. 4.000. Abbonamento annuo (11 fascicoli) L. 40.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 80.000
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizioni: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: La Litografica S.r.l. - Busto Arsizio (VA)
Registrazione: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70% - **Distrib:** MePe, via G. Carcano 32 - Milano

Amarcord

Quando una persona viene a casa mia per la prima volta non può fare a meno di esprimere stupore non appena la introduco in quella che io chiamo "Stanza delle meraviglie":

Su due tavoli trovano posto, infatti, diversi calcolatori, ognuno dei quali accessoriato in maniera pressochè completa; senza parlare delle librerie, piene zeppe di libri e riviste di informatica e di elettronica, o dei contenitori per dischi che mi auguro di poter sostituire al più presto con Compact Disk.

Inevitabile, quindi, la domanda dell'ospite: "Come ha avuto inizio tutto questo?"; altrettanto inevitabile l'invito a sedersi in poltrona per sorseggiare un bicchiere di liquore mentre racconto la "storia" di ogni calcolatore e periferica presente nella stanza.

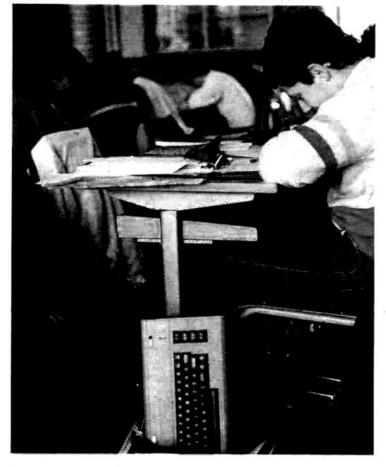
Non saprò mai se la mancanza di interruzione del racconto è dovuta a reale interesse oppure alla raffinata educazione dei miei ospiti; certo è che ricordo con piacere l'acquisto dei primi libri di informatica, le lezioni sui calcolatori che seguivo alla facoltà di Ingegneria al punto da trascurare gli esami "fondamentali".

E non posso fare a meno di ricordare quella persona sconosciuta che, senza volerlo, mi aprì la strada che ora percorro con tanta soddisfazione. Era il maggio del 1980 e, dopo aver visitato l'EDP USA, alla Fiera di Milano, aspettavo l'arrivo della metropolitana e, nell'attesa, leggevo i depliant pubblicitari raccolti in Fiera. A causa delle scarse finanze allora disponibili, ero infatti costretto a limitare le mie pretese acquistando un piccolo calcolatore francese in scatola di montaggio che, tuttavia, costava quasi un milione.

Mentre confrontavo tabelle, prestazioni, prezzi, possibilità di espansione ed altro, mi si avvicinò un signore che aveva visitato la Fiera con lo scopo di rintracciare computer idonei alla trasmissione dati via radio. Era infatti un radioamatore interessato ad ampliare il campo delle sue conoscenze.

Parlando del più e del meno, e venuto a sapere che intendevo spendere non più di un milione di lire, mi suggerì di tornare in Fiera il giorno dopo perchè la Harden (che allora era l'importatrice esclusiva dei prodotti Commodore) aveva diminuito brutalmente il prezzo di vendita del modello PET 2001 che avevo, mio malgrado, escluso dalle scelte possibili a causa del prezzo elevato (prima del maggio '80 costava infatti 1600000 lire).

E tutto il resto, vale a dire la conoscenza del dottor Di Pisa, la collaborazione alla rivista Micro & Personal Computer, la realizzazione di questo, favoloso "Commodore Computer Club" è stato possibile grazie al glorioso PET 2001, passato, in seguito, a miglior vita.

Ora mi capita spesso di entrare nelle aule in cui si insegna informatica: i ragazzi, tutti di età intorno ai sedici

anni, mi fanno tornare alla mente le mie prime digitazioni, i primi (e tanti!) errori di battitura, la rabbia nel non vedere girare i programmi come avrei voluto, l'incrollabile certezza di aver acquistato un computer difettoso.

Non posso fare a meno di sorridere nel vedere la tangibile delusione di questi ragazzi quando, tanto per dirne una, non riescono a rintracciare l'errore dovuto alla mancanza di un FOR oppure ad un RETURN di troppo.

E un po' di malinconia mi assale ricordando quella sera in cui, digitando e dimenticando lo scorrer delle ore, notai all'improvviso un certo chiarore nella stanza: era l'alba che, sopraggiunta senza che me ne accorgessi, sembrava salutare, finalmente, le mie "scoperte" sulla corretta gestione dei file.

Mi capita spesso, ora, di parlare con persone super esperte, che sanno tutto sul Commodore 64 o sulle procedure di invio dati o su tutti i calcolatori, indistintamente. Non posso fare a meno di ammirarle e di invidiarne la notevole preparazione, soprattutto se tengo conto della loro età, troppo spesso nettamente inferiore alla mia.

E ricordo, tuttavia, le telefonate e le lettere di complimenti di quei lettori che, risolti i loro problemi grazie alla lettura dei miei articoli, vogliono, semplicemente, ringraziarmi.

Mi capita spesso, però, di vedere giovani che perdono le loro giornate nei bar, davanti ai videogame, e adesso, grazie all'home computer, anche a casa propria, nell'illusione di sentirsi "grandi" nei confronti dei compagni che totalizzano punteggi minori.

Ecco perchè, in questo numero, non posso fare a meno di pensare ai principianti; a coloro, soprattutto, che in occasione dello scorso Natale hanno ricevuto il loro primo computer. Tra di loro c'è chi si limiterà a giocare, perdendo la straordinaria occasione che gli viene offerta; ma tra i principianti vi sono certamente anche coloro che intendono andare oltre, sia a livello hobbistico che professionale.

Chi ci legge, quindi, non sia severo nel giudicare superficialmente i semplici listati pubblicati che, comunque, si prestano benissimo per miglioramenti di ogni tipo.

E se poi gli "esperti" che ci seguono ripenseranno ai tempi in cui anche loro erano principianti, vale a dire ai veri tempi "duri", sorrideranno di certo nel rievocare le enormi difficoltà rivelatesi poi, fortunatamente, semplici bolle di sapone.

Alessandro de Simone

PER FORZA! NON PARLI LA SUA LINGUA!

Finalmente un libro di circa 400 pagine diverso dagli altri sinora usciti, un libro che fa capire come funziona veramente il tuo Commodore 64 o 128.

Butta via l'assembler, con tutto il suo codazzo di numeri esadecimali e sigle pseudo-mnemoniche! Impara anche tu

IL VERO LINGUAGGIO MACCHINA DEL COMMODORE 64

quello espresso da soli numeri, ciascuno dei quali ha un significato ben preciso. La lingua del Commodore 64 è formata da 151 numeri, di cui solo una ventina frequentemente usati e questo libro ti insegna il significato e l'uso di ciascuno di essi con centinaia di esempi che potrai immediatamente provare direttamente sul tuo Commodore senza alcuna particolare conoscenza o dispositivo.
Ti accorgerai quanto sia facile programmare direttamente in linguaggio macchina senza far ricorso ad ausili strani che finiscono solo per creare una gran confusione. Il libro contiene anche centinaia di routine per le più varie applicazioni: animazione, grafica etc. e contiene numerose tabelle di estrema utilità.

Per ricevere il libro inviare un vaglia postale, un vaglia telegrafico o un assegno bancario di Lire 30.000 comprensive di IVA e spese postali, intestato a:
Società Editrice «Linguaggio Macchina» s.a.s. c/o Studi Professionali Centralizzati, Corso Garibaldi, 95 - 82100 Benevento.

Un'istruzione poco usata

□ Vorrei sapere a che serve l'istruzione WAIT
(Andrea Cammi, Bellona)

• Wait, in inglese, significa "Attendi" ed è appunto il comando da impartire al computer in determinate circostanze.

Un esemipo del comando Wait, che ha tre parametri, di cui l'ultimo opzionale, può essere il seguente:

```
WAIT 197,1
```

in tal modo si arresta l'esecuzione del programma Basic finchè la locazione 197 non assume il valore 1.

Il byte 197 non è stato preso a casaccio: questo contiene, in ogni istante, il codice del tasto eventualmente premuto. Se nessun tasto è premuto, il valore della locazione 197 è uguale a 64. Ti puoi render conto di quanto asserito digitando il mini programma che segue:

```
100 PRINT PEEK(197): GOTO 100
```

Facendolo girare noterai la visualizzazione continua del numero 64 che cambia, però, a seconda dei tasti eventualmente premuti.

In particolare verrà visualizzato il valore unitario se terrai premuto il tasto Return.

Sembrerebbe che l'istruzione Wait sia in grado di sostituire cicli di attesa strutturati ricorrendo all'arcinoto Get:

```
100 GET A$: IF A$="" THEN 100
```

Purtroppo la sostituzione di un ciclo di Get non è così semplice come può sembrare a prima vista e vedremo ora il perchè. Prima, però, è indispensabile chiarire che cosa si intende per OR Esclusivo.

Con tale termine si intende l'operazione logica, effettuata tra due bit, che fornisce come risultato zero, se i due bit sono eguali tra loro (cioè entrambi "1" opppure entrambi "0") mentre il risultatò vale uno se i due bit sono diversi tra loro (cioè uno vale zero e l'altro vale uno).

Riprendiamo ora il discorso ricordando che l'istruzione Wait ha tre parametri che possiamo schematizzare nel modo seguente:

```
WAIT IN,DA1,DA2
```

Il parametro "IN" rappresenta, come già detto, l'indirizzo del byte che vogliamo esaminare. Quando il computer incontra l'istruzione corredata dai tre parametri, effettua un'operazione di OR Esclusivo tra il dato DA2 ed il contenuto della locazione IN; subito dopo, effettua un AND tra il risultato ottenuto e il dato DA1. Se il risultato di quest'ultima operazione è nullo, allora ripete nuovamente l'intero ciclo sospendendo, di fatto, l'esecuzione del programma; se, invece, il risultato è diverso da zero (cioè, bada bene, assume un QUALSIASI valore diverso da zero) il programma Basic continua con l'istruzione successiva.

Se fai girare il seguente programmino

```
100 REM ESEMPIO DI WAIT
110 :
120 REM TASTO RETURN=1
130 REM TASTO "F"=21
140 REM TASTO "R"=17
150 REM TASTO "D"=18
160 FORI=1TO100:NEXT
170 PRINT"PREMI IL TASTO"
180 PRINT"PER IL TEST"
190 WAIT 197,1,64
200 PRINT PEEK(197)
```

ti accorgerai che è piuttosto difficile individuare il valore da assegnare ai due parametri per fare in modo che il programma continui soltanto quando si preme il tasto Return (o qualsiasi altro tasto che desideri). E ciò è dovuto proprio perchè la prosecuzione del programma è decisa soltanto dal fatto che il risultato sia, o meno, eguale a zero.

Un altro motivo per cui non può sostituire del tutto un ciclo Get è dovuto al fatto che non è possibile individuare quale tasto è stato premuto ed effettuare, di conseguenza, un eventuale salto "ragionato" ad altri segmenti di programma.

Il comando Wait, usato con due soli parametri, si rivela utile quasi esclusivamente per fermare un programma finchè non si preme un tasto qualunque.

Un buon carattere

□ Vorrei sapere come posso visualizzare 80 colonne con il mio C/64, come stampare in minuscolo su stampante e come raddoppiare la grandezza dei caratteri.
(Giuseppe Buccolieri, S. Pancrazio)

• Vi sono in giro alcune routine, di cui abbiamo già parlato, che consentono di visualizzare 80 colonne invece delle solite 40. Ricordiamo, però, che non possono essere usate unitamente a programmi "professionali" (come Word Processor, Data Base) o, comunque, protetti, dal momento che la routine stessa viene in genere automaticamente esclusa non appena si caricano i programmi in oggetto. Inoltre la visualizzazione delle 80 colonne è accettabile solo su monitor in bianco e nero; su TV a colori, ad esempio, è un vero e proprio disastro e stanca subito la vista.

Per stampare su carta caratteri minuscoli e maiuscoli (anzichè maiuscoli e semigrafici) è necessario assegnare il terzo parametro al comando di apertura del file. Esempio:

```
OPEN 4,4,7: CMD4: LIST
```

In questo modo potrai stampare un listato Basic nel modo richiesto che prende, più propriamente, il nome di "Business Mode".

Per stampare caratteri ingranditi, ti ricordiamo che è possibile soltanto raddoppiarne la larghezza (e non l'altezza), facendo precedere il messaggio da stampare dal carattere "speciale" CHR$(14). Esempio:

```
OPEN 4,4: PRINT CHR$(14)"ESEMPIO"
```

Alcune stampanti, come la STAR NL/10, consentono anche il raddoppio in altezza dei caratteri.

Un personaggio in cerca di editore

□ Vorrei sapere come mettere in vendita del software da me scritto. E' necessaria qualche autorizzazione per usare nomi di personaggi tratti da celebri film?
(Antonio Fiorito, Torremaggiore)

• Per porre in vendita qualsiasi cosa è necessario essere iscritti alla Camera di Commercio, avere partita IVA e sbrigare altre pratiche burocratiche che ti può spiegare in dettaglio un commercialista

di tua fiducia al quale ti consigliamo di rivolgerti prima di fare qualunque passo.

Per ricevere compensi da riviste del settore, invece, la procedura è molto più semplice dal momento che ti viene regolarmente rilasciata una ricevuta attestante il compenso ricevuto per una prestazione saltuaria.

La seconda parte della domanda è di notevole interesse e mette in luce un fenomeno legato al consumismo che prende il nome di Merchandising.

In pratica si tratta di questo: non appena un film, un personaggio o, comunque, un evento legato prevalentemente al mondo dello spettacolo, ha successo, subito si cerca di far quattrini sfruttando la notorietà del momento. Le magliette, i quaderni, i dischi, i profumi legati, ad esempio, a Rambo, a Snoopy a E.T. e ad altri personaggi, devono pagare diritti di "immagine" (spesso molto salati) agli autori o a coloro che, comunque, detengono i "diritti" d'immagine. La IBM ha pagato una certa cifra per utilizzare l'immagine di Charlot, come pure la Mondadori, per i suoi simpatici cartoncini di auguri, paga una quota per riprodurre i personaggi di Schulz (Snoopy, Charlie Brown ed altri).

Anche le software house non sono da meno e pagano i diritti per poter assegnare nomi di richiamo ai videogiochi che immettono sul mercato (Aliens, Rambo, Dynasty eccetera).

Fatevi i dischi vostri

□ E' possibile leggere con il 1541 i dischetti formattati con altri drive?
(Paolo Costabel, Genova)

• Un disco vergine è, in pratica, simile ad un immenso foglio di carta bianca sul quale si può scrivere in un'inifinità di modi.

Poichè il computer ha bisogno del massimo ordine per scrivere (e leggere) dati, si procede all'operazione di "formattazione" che, in pratica, provvede a scrivere tanti quadretti ben ordinati secondo un allineamento che è tipico di ogni computer.

Ogni disco, dopo la formattazione, assomiglia ad un quaderno; come tu sai, però, esistono molti tipi di quaderni che, pur essendo identici all'esterno, non lo sono all'interno a causa del diverso formato (e quantità) dei fogli, a quadretti, a righe, a righe large, medie, strette, commerciali eccetera.

Mentre è per noi possibile scrivere esercizi di matematica anche su un quaderno a righe (benchè l'ideale sia uno a quadretti) un computer non è in grado di scrivere (o di leggere) da un quaderno al quale non è... abituato.

In conclusione ogni computer ha il suo formato. Alcuni, però, vantano la cosiddetta compatibilità per cui sono in grado di accettare dischi provenienti da altri tipi di computer.

Nel caso dei prodotti Commodore non esistono in commercio drive di altri calcolatori in grado di esser letti dal tuo 1541. Anche se riuscissi nell'intento, però, è probabile che i file letti non ti siano di alcuna utilità a causa della notevole diversità dei codici esistente tra i vari computer sul mercato.

Caccia all'errore

□ Digitando listati da altri libri (o riviste del settore) sorgono spesso errori o malfunzionamenti di difficile interpretazione. Quale può esser la causa?
(Salvatore Caramanna e altri lettori)

• Purtroppo non siamo in grado di valutare la qualità del software della cosiddetta "concorrenza" e, benchè sia significativo il fatto che vi rivolgiate a noi anzichè agli autori del software in questione, ci permettiamo tuttavia di spezzare una lancia in favore dei listati pubblicati su periodici... cartacei (come il nostro).

L'esperienza ormai ha dimostrato che i listati "rigorosamente controllati", come affermano i lettori che li hanno trascritti, non sono per nulla privi di errori di digitazione. Controllate, quindi, con il massimo rigore le linee Basic (UNA PER UNA) attenedovi alle raccomandazioni che, quasi sempre, accompagnano gli articoli a corredo dei programmi.

Effetti speciali

□ Provate ad effettuare, in sequenza, le seguenti operazioni: 1- spegnete ed accendete il C/64; 2- digitate SYS 63490; 3- premete dieci volte il tasto Crsr Dwn; 4- digitate qualche carattere a caso; 5- tenete premuto il tasto Crsr Dwn ed osservate l'effetto che ne consegue. Perchè accade?
(Davide Crivellotti, Mediglia)

• Mah! probabilmente perchè viene alterato il puntatore alla posizione del cursore sullo schermo. Ma toglimi una curiosità: come diavolo ti sei accorto dello strano fenomeno?

Numeri e parole

□ Come si possono leggere DATA contenenti valori numerici e parole?
(Cristian Di Paolo, Roma)

• Digita attentamente il seguente programma:

```
100 FORI=1 TO 9:READ A$
105 IFVAL(A$)=0 THENPRINT"QUESTA E' UNA PAROLA "A$:GOTO130
110 PRINT"QUESTO E' UN NUMERO "A$:GOTO130
120 DATA PIPPO,GATTO,23,-98,GENNAIO,90,-.98,CRISTIAN,0
130 NEXT
```

Come puoi notare l'istruzione Read è in grado di leggere indifferentemente numeri o parole. Se non puoi stabilire, prima della lettura tramite Read, se il prossimo Data da leggere sarà un numero oppure un nome, è sufficiente avere l'accortezza di imporre la lettura di una stringa, vale a dire "Read A$" e non "Read A".

Non appena leggi A$ puoi "estrarne" l'eventuale valore numerico con Val(A$): se questo è nullo, è quasi certamente un nome, altrimenti è un valore numerico. Se tentassi di sostituire A$ con A (riga 100), otteresti un "Syntax Error in 120" dal momento che il computer riceve il comando di leggere un numero (A) e trova, al contrario, una stringa (Pippo).

Se il valore di A$ è nullo, non è detto però che sia un nome: lo dimostra l'ultimo messaggio visualizzato facendo girare il programma di prima. Per esser sicuri di distinguere esattamente una parola da un numero è necessario introdurre dei "filtri" software in grado di accorgersi della differenza. Prova, infatti, ad inserire la seguente riga-filtro:

```
102 IF A$="0" THEN 110
```

e vedrai che l'ultimo messaggio sarà più coerente di quello visualizzato in precedenza.

La mappa della memoria

□ Vorrei proporvi di pubblicare una tabella con l'indicazione delle locazioni di memoria del C/64 ed il loro utilizzo.

Inoltre vorrei sapere come pulire le testine del drive senza l'uso dell'apposito pulisci testine.
(Lorenzo Rossi, Rimini)

• Ciò che tu chiedi è la cosiddetta mappa della memoria la cui pubblicazione, commentata, richiederebbe un intero libro limitato, per di più, ai soli esperti.

A che serve dire, infatti, che le locazioni 122/123 rappresentano il puntatore del byte corrente nel testo Basic (come si li-

mita, appunto, a riferire la "Guida di riferimento per il C/64") se non si scrive un intero articolo corredato di programmi per illustrarne possibili usi?

Ma, a pensarci bene, è proprio ciò che noi facciamo articolo per articolo, numero per numero di Commodore Computer Club!

Invece di spiegare i parametri di una particolare istruzione grafica (o una "strana" Poke), preferiamo pubblicare un programma che ne faccia uso, in modo che il lettore, se realmente interessato, può proseguire le ricerche per proprio conto.

In quasi tutti gli articoli che pubblichiamo c'è sempre qualche riferimento a particolari locazioni di memoria, o ad usi insoliti di comandi Basic, di certo molto più utili di aride tabelle non finalizzate.

Se, comunque, possiedi un C/128, puoi procurarti i numeri della rivista "Personal Computer" (a partire dal N.6) che pubblica a puntate l'intera mappa della memoria del C/128. Non pensare, però, che si impari molto avendo a disposizione solo la mappa: l'unico sistema per realizzare qualcosa di valido consiste (purtroppo) nel lavorare tantissimo e sperimentare tecniche non "ufficiali".

Per ciò che riguarda la seconda parte della domanda non capisco il motivo di tale desiderio di pulizia. Posseggo più di un drive da molto tempo, li uso in media quattro ore al giorno e non mi sono mai sognato di aprirli per questioni di sporcizia; se, a mia insaputa, all'interno dei drive ha preso posto una tribù di pipistrelli o di gatti selvatici, il loro ecosistema non pregiudica la funzionalità del drive.

Scherzi a parte, se hai l'accortezza di coprire con un panno le fessure del drive quando non lo usi, e se provvedi a custodire con cura i dischetti, non dovrebbero insorgere inconvenienti di sorta per il buon funzionamento della periferica.

Il nuovo Drive 1541-C e lo Speed-Dos

□ Che differenza c'è tra il vecchio 1541 (custodia marrone) ed il nuovo 1541-C (custodia bianca)?
(Numerosi lettori)

• Da un punto di vista software, nessuna (almeno così assicura chi lo usa da diversi mesi). Da un punto di vista Hardware, però, vi sono piccole ma sostanziali differenze nella disposizione dei componenti sul circuito stampato ed è necessario tenerne conto nel caso si desideri apportare variazioni.

Una di queste è sicuramente lo Speed Dos, un accessorio H/W che velocizza le operazioni di trasferimento dati, di cui abbiamo parlato tempo fa.

Ne approfitto, anzi, per ricordare che la ditta Niwa di Sesto San Giovanni (tel.02-24.40.77.6) è costretta ad allungare i tempi di consegna del comodo accessorio (dal prezzo di L.70000 circa) perchè i richiedenti dimenticano di precisare il tipo di drive in loro possesso: siate chiari, quindi, quando ordinate accessori per il vostro disk drive!

Totocalcio illeggibile

□ Sul N.36 di CCC alcune righe del listato "Archivio Totocalcio" risultano illeggibili (segue elenco). Potete ripubblicarle?
(Raffaele Esposito, Roma, ed altri lettori)

• Siamo davvero mortificati per il fatto che la rubrica Totocalcio sembra esser penalizzata per un motivo o per un altro! Ci risulta, fortunatamente, che solo una minima percentuale di copie del N.36 siano stampate in modo incompleto. Evidentemente alla nostra tipografia manca un controllo computerizzato in tempo reale...

```
 20 GET A$:IF A$-"S" THEN 50
 30 IF A$-"N" THEN LIST 650:PRI
    NT TAB(10)"[UP]↑":LIST 660:
    PRINT TAB(10)"[UP]↑":END
 40 GOTO 20
200 GET A$:IF A$-"E" THEN A-914
    4:POKE 8915,184:POKE 8916,3
    5:GOTO 900
210 IF A$-"C" THEN 230
250 A-PEEK(8915)+256*PEEK(8916)
    :PRINT"[DOWN]COLONNE INSERI
    TE:";(A-9144)/13:B-800
260 IF PEEK(8926)-1 THEN B-500
270 PRINT"[DOWN]PUOI INSERIRNE:
    "B-(A-9144)/13
```

Compatibilità C/64 C/128

□ Mi hanno detto che per alcuni giochi per C/64, che non funzionano sul C/128 predisposto in modo 64, è necessario cambiare il contenuto di una locazione prima di dare il Run. E' vero?
(Daniele Alfonso, Alghero)

• Ho sentito anch'io parlare di alcune (presunte) incompatibilità del C/128 in modo 64. Forse la locazione alla quale fai cenno è la N.1 che viene automaticamente modificata tenendo premuto il tasto Caps Lock. Posseggo un paio di programmi di utilità, infatti, che non vogliono saperne di funzionare se dimentico sollevato quel tasto. Altre incompatibilità forse ve ne sono ma, finora, nessuno ha affrontato il problema in modo sistematico per individuare i limiti del modo 64.

Se, naturalmente, qualche lettore ha condotto studi in proposito, saremo ben lieti di pubblicare il resoconto delle sue esperienze.

Ipotesi non azzardate

□ Premetto che lascio sempre inserito il dischetto nel drive sia quando accendo che quando spengo. Da un po' di tempo, però, mi accorgo che alcuni file non campaiono più nella directory nonostante io non li abbia espressamente cancellati.
(M. Evangelisti, Modena)

• Se l'apparecchio al quale ti riferisci quando dici "accendo e spengo" è il drive, mi meraviglio che vi sia ancora qualche file sul disco e non che ne manchino alcuni. Se invece ti riferisci al computer, nessuna preoccupazione: tutte le volte che accendi il computer viene inviato un segnale di reset anche al drive che non fa altro che re-inizializzarsi.

Il dischetto va inserito solo quando il drive è acceso e tolto prima di spegnerlo. Fa eccezione il caso in cui si desidera attivare la procedura di Bootstrap che, per il PC-10 e per il C/128 in modo 128 (oppure CP/M) richiedono espressamente l'inserimento del dischetto prima dell'accensione del drive. Ti consigliamo, però, di seguire tale procedura solo se possiedi un 1571, predisposto per il Bootstrap. Se hai un 1541, accendilo, inserisci il disco e, in seguito, accendi il computer per il caricamento automatico.

Se ritieni di aver sempre seguito scrupolosamente le indicazioni appena dette, l'unica spiegazione logica che giustifica la cancellazione di alcuni file è dovuta al fatto che utilizzi la dannata chiocciolina per registrare un file dotato dello stesso nome di un altro file presente su disco: per un po' di tempo tale forma sintattica "funziona" correttamente ma capita almeno una volta che la BAM va in tilt, rovinando numerose zone del dischetto.

Per registrare un nuovo file, pertanto, ricordarsi di attenersi rigorosamente alla seguente procedura:

• Registrare il nuovo file (sequenziale, programma, relativo o user che sia) con un nome di comodo (es. PIPPO).
• Cancellare il vecchio file.
• Cambiar nome a PIPPO con il comando di Rename.

Credimi: è l'unico sitema per evitar sorprese amarissime che fanno malissimo.

Videogames per Amiga

Una breve panoramica sul fronte dei videogiochi per il tuo nuovo, fiammante computer

di Giovanni Valli e Ornella Fidani

Come abbiamo già detto la volta scorsa, il Commodore Amiga è un computer che può essere utilizzato in molteplici campi e quasi sempre con buoni risultati. Questa volta ci riferiremo al campo "disimpegnato" dei giochi. Nostro compito, infatti, è il verificare se le notevoli possibilità dell'Amiga vengano realmente sfruttate dal software per ora disponibile.

Con la nostra esperienza possiamo affermare che, quando viene immesso sul mercato un nuovo computer, non tutti i giochi reperibili possono essere subito di buon livello. Non diementichiamo, infatti, i primi videogame del C/64 che erano a malapena dello stesso livello del Vic-20. A volte, inoltre, alcuni giochi non sono altro che il frutto di modesti adattamenti di giochi creati in precedenza per altri computer (nella fattispecie, il C/64).

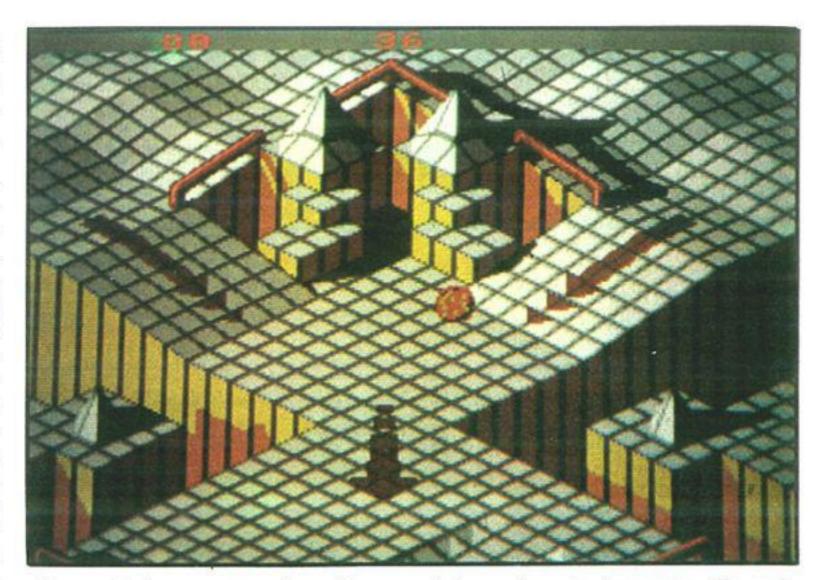
Iniziamo a parlare del primo programma che ci è capitato di esaminare, in ordine di tempo, che si chiama **"OTHELLO"** ed è scritto semplicemente in Basic da David Addison.Non è altro che la versione per l'Amiga del noto gioco da tavolo che si svolge fra due avversari, dei quali uno dei due può essere il computer. Bisogna dire che la schermata grafica è ben realizzata e, a vedersi, è anche riposante.

Senza dilungarci sulle regole del gioco, che tutti conosciamo, facciamo notare soltanto che la scacchiera è ben raffigurata in tre dimensioni, ma che i comandi per posizionare le pedine (uno per direzione) sono molto scomodi da usare. A nostro avviso sarebbe stato meglio utilizzare il mouse, fornito di serie con la macchina, per rendere più comodo lo spostamento delle pedine.

Il secondo gioco che ha attratto la nostra attenzione è **"DELTA PATROL"**, prodotto dalla Other Valley Software. In questo game, un elicottero sorvola una landa deserta, cimentandosi contro nemici volanti; il velivolo, però, deve atterrare spesso per rifornirsi di carburante. Come si può chiaramente capire, questo gioco

non è certo innovativo e rammenta uno dei programmi del C-64, del quale ricalca anche la grafica. Di certo si poteva fare di più e sicuramente saranno proposte, nel prossimo futuro, altre versioni di maggior effetto.

Vogliamo ora parlare di **"MARBLE MADNESS"**, versione per l'Amiga del celebre gioco che furoreggiava nei Bar un paio di anni fa, composto da più fasi da superare in un tempo ben preciso. Il giocatore deve condurre all'arrivo, in ogni fase, una sfera rossa attraverso un cammino irto di ostacoli, cercando di incanalarla in stretti passaggi e fronteggiando ripetuti attacchi di nemici implacabili; questi sono rappresentati da un gruppo di bruchi che, tra l'altro, fanno addirittura capriole per non far scappar via la preda; le lanciano contro acidi corrosivi, martelli ed aspiratori che compaiono all'improvviso, ponti mobili e perfino una gigantesca onda alla fine della terza fase.

A complicare la faccenda si aggiunge il fatto che non è semplice manovrare il mouse, tenendo conto della prospettiva e della pendenza dei vari percorsi. Da quanto detto si comprenderà che questo è un gioco molto impegnativo e coinvolgente, al punto da rimanere incollati allo schermo per lungo tempo. A ciò contribuisce anche il sottofondo musicale, diverso per ciascuna fase del gioco, che sottolinea i numerosi momenti di tensione.

In conclusione non possiamo fare a meno di ammettere che la grafica è ottima e curata nei minimi dettagli, tanto da fugare gli ultimi dubbi sulle reali possibilità grafiche.

Questa prima panoramica sui programmi da svago termina con un cenno a **"MEAN 18"**, edito da Accolade. In questo videogame ci troviamo in un campo da golf, ben realizzato e facile da percorrere. Sottolineamo che gli autori hanno cercato, riuscendo in pieno con successo, di ricreare situazioni reali (per quanto riguarda, ad esempio, la dinamica del movimento impresso alla pallina).

Quando si fa partire il programma, la prima schermata mostra una simpatica talpa che sbuca dal suolo; successivamente vengono poste numerose domande sulla scelta dei campi, sul numero e sull'identità dei giocatori, sulla intenzione o meno di far pratica prima dell'inizio del gioco stesso. Dalla posizione di partenza, dopo aver osservato in pianta il campo relativo alla prima buca, appare l'immagine del campo stesso vista dalle spalle del giocatore; di fronte, in prospettiva, la buca. Si ha una chiara visione degli alberi, dei tipi di fondo, e poco importa se la risoluzione grafica non è delle migliori perchè tutti i particolari salienti sono ben rappresentati. Sul bordo inferiore del video troviamo un gruppo di informazioni e di comandi (tra cui, ad esempio, quello per cambiare la direzione del tiro). All'estrema sinistra dello schermo una colonnina graduata indica la forza da imprimere, che si sprigiona nel momento in cui si schiaccia il pulsante del mouse. Quando si arriva al GREEN, il punto di vista si sposta al di sopra della testa del giocatore e viene indicata con una linea la direzione del tiro.

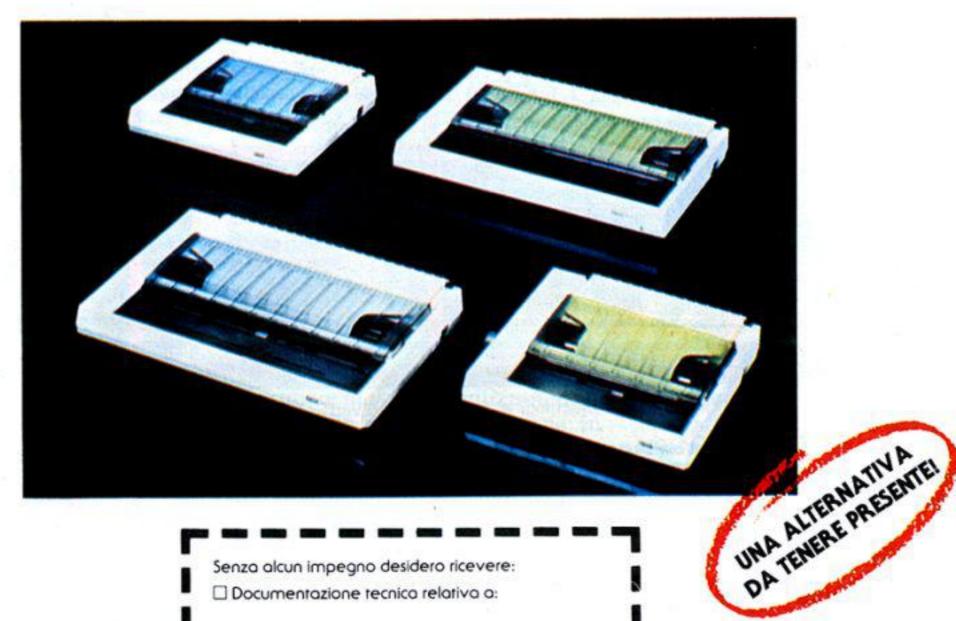
Notevoli sono gli effetti sonori e i rumori che accompagnano il colpo ed il conseguente movimento della pallina, davvero molto realistici.

TELCOM IMPORTA E DISTRIBUISCE LE STAMPANTI CITIZEN

Una gamma di stampanti che copre tutte le esigenze:
- stampa alfanumerica normale, ascendente-discendente, espansa, compressa, Elite, Near Letter Quality, grafica compatibile IBM, 8 Kbytes di buffer di stampa, trattore di moduli continui, inseritore di fogli singoli, testina a 9 aghi (24 aghi su HQP 45).

MODELLO	NR. COLONNE	VELOCITÀ (CAR/SEC)
LSP 10	80	120
MSP 10	80	160
MSP 15	136	160
MSP 20	80	200
MSP 25	136	200
HQP 45	136	200

E per gli hobbisti... la piccola 120D: 80 colonne, 120 caratteri car/sec con interfaccia parallela, seriale, compatibile **APPLE, COMMODORE.**

UNA ALTERNATIVA DA TENERE PRESENTE!

Senza alcun impegno desidero ricevere:
☐ Documentazione tecnica relativa a:
___
☐ Visita di un Vs. Funzionario
Nome e Cognome ___
Società/Ente ___
Via ___
Città ___ CAP ___
Telefono ___
CCC

Telcom srl - 20148 Milano - Via M. Civitali, 75
Tel. 02/4047648-4049046
Telex 335654 TELCOM I - Telefax 02/437964

adverteam

La Grande Libreria Systems

Autori Vari

64 Programmi per Commodore 64

Giochi, grafica, gestione delle stringhe, musica, numeri, gestionali.

Lire 4.800

Autori Vari

I miei amici C16 & Plus4

Un manuale pratico per padroneggiare il basic di questi computer.

Lire 7.000

Autori Vari

Strategie vincenti per Commodore 64

Le strategie per tutti i classici del videogioco: per giocarli, vincerli o programmarli.

Lire 5.800

Autori Vari

62 Programmi per il Vic 20, C16 e Plus 4

Giochi, grafica e routine per imparare a programmare.

6.500

Roberto Didoni, Guido Grassi

Utilities e giochi didattici

Raccolta di programmi pratici per tutti i Commodore e lo Spectrum.

Lire 6.500

Giovanni Mellina

Tutti i segreti dello Spectrum

4 passi nella Rom: come usare le più importanti routine del sistema operativo.

Lire 7.000

Roberto Didoni, Guido Grassi

Simulazioni e test per la didattica

Teoria e listati per Vic 20, C16, C64 C128 e Spectrum Sinclair.

Lire 7.000

Paolo Goglio

Impara giocando il basic dello Spectrum

Esercizi pratici per entrare nel vivo della programmazione.

Lire 7.000

Clizio Merli

μPascal per Commodore 64/128

Un manuale completo per il programma compilatore

Lire 7.000

Umberto Colapicchioni e Luca Galuzzi

Dal registratore al drive del C64

Tutti i segreti delle memorie di massa del Commodore 64

Lire 7.000

Autori Vari

ADA

Il linguaggio passepartout dei computer degli anni '80.

Lire 5.000

Clizio Merli

Il linguaggio PASCAL

Un manuale tascabile per lo studio e la programmazione.

Lire 5.000

Sì, voglio arricchire la mia biblioteca con i seguenti volumi al prezzo di copertina + lire 3.000 per spese di spedizione.

- ☐ **64 Programmi per Commodore 64**
- ☐ **Strategie vincenti per i tuoi videogames**
- ☐ **62 Programmi per Vic 20 C16 e Plus77**
- ☐ **Utilities e giochi didattici**
- ☐ **Tutti i segreti dello Spectrum**
- ☐ **Simulazioni e test per la didattica**
- ☐ **Imparare giocando il basic dello Spectrum**
- ☐ **I miei amici C16 e Plus4**
- ☐ **Pascal per Commodore 128**
- ☐ **Dal registrattore al drive del C64**
- ☐ **ADA**
- ☐ **Il linguaggio Pascal**

Nome ....

via .... **N.ro.** .... **telefono** ....

CAP .... **Città** ....

Su tale importo mi praticherete lo sconto del 10% in quanto abbonato a ☐ *Commodore Computer Club* ☐ *Personal Computer* ☐ *Com puter* ☐ *VR Videoregistrare.* Pertanto vi invio la somma soltanto di lire ....

Valore dell'ordine lire..................

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Koala per C/16

Un breve programma che consente, tuttavia, di sfruttare le potenzialità grafiche del piccolo computer

di Marco Saetta

Con questo programma, gestibile con joystick e tastiera, si possono ottenere simpatici disegni in alta risoluzione, grazie al ricorso intensivo alle istruzioni grafiche.

Chi possiede il C/16 non deve digitare le REM nè lasciare spazi tra istruzioni successive dello stesso rigo; se, pertanto, viene segnalato un "Out Of Memory Error" vuol dire che non vi siete attenuti al consiglio suggerito.

I disegni eseguiti possono esser registrati su disco oppure su nastro. Chi possiede il registratore deve digitare la riga 535 (e non la 530), mentre i possessori di disk drive trascriveranno la 530 (e non la 535).

Un C/16, come è noto, ha a disposizione una quantità molto limitata di memoria, specialmente se si utilizza la grafica in alta risoluzione. Non è quindi stato possibile inserire istruzioni che "restituissero" il programma una volta registrata la schermata mediante la manipolazione del buffer di tastiera e il Monitor di linguaggio macchina.

Dopo aver caricato una schermata, invece, è possibile recuperare il programma digitando "alla cieca" (senza, cioè, che possiate vedere ciò che battete) il comando diretto:

GOTO 30

I possessori di C/128 devono modificare in parte il programma che, a causa di alcune POKE, non è completamente compatibile. Il listato non è ottimizzato ed è possibile risparmiare molti byte, utilizzando altre tecniche di programmazione, in un modo da inserire altri comandi in grado di rendere ancora più versatile il programma stesso.

Come usare il programma

Inserire il Joystick nella porta N.1 per spostarsi nelle varie direzioni.
Premere Fire, mentre vi spostate, per disegnare.
Tasto "1" per cambiare colore al pennello (max. tre colori diversi).
Tasto "2" per memorizzare le coordinate del punto (X) indicato dal Joy in un particolare momento.
HOME per tracciare un segmento dal punto in cui vi trovate (Y) al punto (X) fissato precedentemente.
ESC per cancellare la riga.
RETURN per cancellare lo schermo.
SPAZIO per colorare un'area delimitata.
C per disegnare un cerchio (vengono chiesti i parametri).
L per caricare un disegno.
* per registrare il disegno effettuato.
8 per definire il punto "attuale" come vertice di un rettangolo.
B per tracciare il rettangolo.
= per cancellare un rettangolo.
INST per cancellare un cerchio.
CRSR per spostare il pennello velocemente, senza lasciar traccia.

```
10 COLOR0,2,3:COLOR4,2,2:COLOR3,1:COLOR1,2,2:COLOR2,2,4:REM DEFINISCE I COLORI
20 GRAPHIC3,1:A=80:S=100:Q=1:P=S:O=A
30 GETA$:J=ASC(A$):IFJ=49THENQ=Q+1:IFQ=4THENQ=1
40 IFJOY(1)=1THENDRAW0,A,S:S=S-1:DRAWQ,A,S:REM JOY ALTO
50 IFJOY(1)=2THENDRAW0,A,S:S=S-1:A=A+1:DRAWQ,A,S:REM JOY DIAG.ALTO DESTRA
60 IFJOY(1)=3THENDRAW0,A,S:A=A+1:DRAWQ,A,S:REM JOY DESTRA
70 IFJOY(1)=4THENDRAW0,A,S:A=A+1:S=S+1:DRAWQ,A,S:REM JOY DIAG.BASSO DESTRA
80 IFJOY(1)=5THENDRAW0,A,S:S=S+1:DRAWQ,A,S:REM JOY BASSO
90 IFJOY(1)=6THENDRAW0,A,S:A=A-1:S=S+1:DRAWQ,A,S:REM JOY DIAG.BASSO SINISTRA
100 IFJOY(1)=7THENDRAW0,A,S:A=A-1:DRAWQ,A,S:REM JOY SINISTRA
110 IFJOY(1)=8THENDRAW0,A,S:A=A-1:S=S-1:DRAWQ,A,S:REM JOY DIAG.ALTO SINISTRA
120 IFJOY(1)=129THENS=S-1:DRAWQ,A,S:REM DISEGNA ALTO
130 IFJOY(1)=130THENA=A+1:S=S-1:DRAWQ,A,S:REM DISEGNA DIAG.ALTO DESTRA
140 IFJOY(1)=131THENA=A+1:DRAWQ,A,S:REM DISEGNA DESTRA
150 IFJOY(1)=132THENA=A+1:S=S+1:DRAWQ,A,S:REM DISEGNA DIAG.BASSO DESTRA
160 IFJOY(1)=133THENS=S+1:DRAWQ,A,S:REM DISEGNA BASSO
170 IFJOY(1)=134THENA=A-1:S=S+1:DRAWQ,A,S:REM DISEGNA DIAG.BASSO SINISTRA
180 IFJOY(1)=135THENA=A-1:DRAWQ,A,S:REM DISEGNA SINISTRA
190 IFJOY(1)=136THENA=A-1:S=S-1:DRAWQ,A,S:REM DISEGNA DIAG.ALTO SINISTRA
200 IFJ=157THENDRAW0,A,S:A=A-10:DRAWQ,A,S:REM FRECCIA SINISTRA
210 IFJ=29THENDRAW0,A,S:A=A+10:DRAWQ,A,S:REM FRECCIA DESTRA
220 IFJ=145THENDRAW0,A,S:S=S-10:DRAWQ,A,S:REM FRECCIA BASSO
230 IFJ=17THENDRAW0,A,S:S=S+10:DRAWQ,A,S:REM FRECCIA ALTO
240 IFJ=32THENA=A+1:S=S+1:PAINTQ,A,S,1:REM COLORA AREA
250 IFJ=13THENGOTO20:REM CANCELLA SCHERMO
260 IFJ=50THENO=A:P=S:REM FISSA PUNTO X
270 IFJ=19THENDRAWQ,A,STOO,P:REM DISEGNA LINEA
280 IFJ=20THENCIRCLE0,A,S,R/2,R:REM CANCELLA CERCHIO
290 IFJ=67THENGOSUB370:REM CERCHIO
300 IFJ=42THENGOTO520:REM SALVA DISEGNO
310 IFJ=76THENGOSUB550:REM CARICA DISEGNO
320 IFJ=56THENL=A:K=S:REM FISSA PUNTO X PER BOX
330 IFJ=66THENBOXQ,L,K,A,S:REM DISEGNA BOX
340 IFJ=61THENBOX0,L,K,A,S:REM CANCELLA BOX
350 IFJ=27THENDRAW0,A,STOO,P:REM CANCELLA LINEA
360 GOTO30:REM CICLO LOOP
370 GRAPHIC0:INPUT"]RAGGIO";R
380 PRINT"■■C-CERCHIO":PRINT"S-SEMICERCHIO":PRINT"A-ARCO DI CIRC."
390 INPUTX$:IFX$="C"THENGRAPHIC3:CIRCLEQ,A,S,R/2,R:RETURN
400 IFX$="S"THENGOTO420
410 IFX$="A"THENGOTO470
420 PRINT"]A-ALTO":PRINT"D-DESTRA":PRINT"B-BASSO":PRINT"Z-SINISTRA"
430 INPUTX$:IFX$="A"THENGRAPHIC3:CIRCLEQ,A,S,R/2,R,270,90:RETURN
440 IFX$="D"THENGRAPHIC3:CIRCLEQ,A,S,R/2,R,360,180:RETURN
450 IFX$="B"THENGRAPHIC3:CIRCLEQ,A,S,R/2,R,90,270:RETURN
460 GRAPHIC3:CIRCLEQ,A,S,R/2,R,180,360:RETURN
470 PRINT"]A-ALTO SIN":PRINT"D-ALTO DES":PRINT"Z-BASSO SIN":PRINT"X-BASSO DES"
480 INPUTX$:IFX$="A"THENGRAPHIC3:CIRCLEQ,A,S,R/2,R,270,360:RETURN
490 IFX$="D"THENGRAPHIC3:CIRCLEQ,A,S,R/2,R,,90:RETURN
500 IFX$="Z"THENGRAPHIC3:CIRCLEQ,A,S,R/2,R,180,270:RETURN
510 GRAPHIC3:CIRCLEQ,A,S,R/2,R,90,180:RETURN
520 GRAPHIC0:INPUT"NOME PROGRAMMA";NM$
525 REM RIGA 530 VERSIONE PER DISK DRIVE
530 PRINT"]M┌":PRINT"S■■■■■■S"+CHR$(34)+NM$+CHR$(34)+"08 2000 3F50":PRINT"■■X"
532 REM RIGA 535 VERSIONE PER NASTRO CASSETTA
535 PRINT"]M┌":PRINT"S■■■■■■S"+CHR$(34)+NM$+CHR$(34)+"01 2000 3F50":PRINT"■■X"
540 POKE239,6:POKE1319,19:FORI=0TO7:POKE1320+I,13:NEXT:END
550 GRAPHIC0:INPUT"NOME PROGRAMMA";NM$
560 GRAPHIC3:PRINT"]M┌":PRINT"S■■■■■■L"+CHR$(34)+NM$+CHR$(34)+"08":PRINT"■■X":PR
INT"■GRAPHIC3"
570 POKE239,6:POKE1319,19:FORI=0TO7:POKE1320+I,13:NEXT
580 END
```

I listati della Systems Editoriale

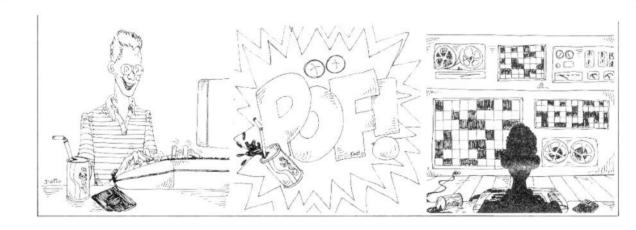
Un'elevata percentuale dei nostri lettori è alle prime armi nel mondo dell'informatica e incontra difficoltà nel digitare i programmi da noi pubblicati.

I caratteri "speciali" bianchi su fondo nero (semi-grafici in "reverse") che rappresentano precisi comandi per i computer Commodore sono riportati nel listato di esempio a sinistra così come appaiono digitandoli su video o su stampante, mentre a destra come li rappresentiamo nei nostri listati.

La riga 360, ad esempio, deve così esser interpretata:
dopo aver battuto il carattere di virgolette (") che si ottiene premendo il tasto SHIFT insieme con il tasto 2, è necessario battere il carattere CRSR DOWN (il tasto, cioè, che normalmente sposterebbe il cursore nella cella video sottostante).

Analogamente, nella riga 180 del listato "tradotto" (di destra), il termine [NERO] sta a significare che bisogna utilizzare il carattere speciale del colore nero (tasto CTRL insieme con il tasto 1, vedi listato di sinistra).

Per ricordare in che modo vengono normalmente visualizzati i caratteri speciali, nella seconda parte delle righe di sinistra (dopo i REM) sono riportati i tasti che è necessario premere per ottenere il carattere-comando "speciale".

La Redazione

```
100 REM I CARATTERI SPECIALI
110 REM DEI COMPUTER COMMODORE
120 REM COME APPAIONO NORMALMENTE
130 REM SU VIDEO O SU CARTA.
140 REM (CTRL = TASTO CONTROL)
150 REM (CMDR = TASTO COMMODORE)
160 REM (CRSR = TASTI CURSORE)
170 :
180 PRINT"■":REM CTRL+1 NERO
190 PRINT"▣":REM  "  +2 BIANCO
200 PRINT"▣":REM  "  +3 ROSSO
210 PRINT"◣":REM  "  +4 AZZURRO
220 PRINT"▩":REM  "  +5 PORPORA
230 PRINT"▣":REM  "  +6 VERDE
240 PRINT"▣":REM  "  +7 BLU
250 PRINT"▣":REM  "  +8 GIALLO
260 PRINT"▣":REM  "  +9 REVERSE ON
270 PRINT"■":REM  "  +0 REVERSE OFF
280 PRINT"▣":REM CMDR+1 ARANCIO
290 PRINT"▣":REM  "  +2 MARRONE
300 PRINT"▣":REM  "  +3 ROSSO CHIARO
310 PRINT"▣":REM  "  +4 GRIGIO 1
320 PRINT"▣":REM  "  +5 GRIGIO 2
330 PRINT"▣":REM  "  +6 VERDE CHIARO
340 PRINT"▣":REM  "  +7 BLU CHIARO
350 PRINT"▣":REM  "  +8 GRIGIO 3
360 PRINT"▣":REM  CRSR IN BASSO
370 PRINT"▣":REM  CRSR A DESTRA
380 PRINT"▣":REM  CRSR IN ALTO
390 PRINT"▣":REM  CRSR SINISTRA
400 PRINT"▣":REM  HOME
410 PRINT"▣":REM  CANCELLA SCHERMO
420 :
430 REM  ESEMPI DI VISUALIZZAZIONE:
440 PRINT"▣▣▣▣▣▣▣":REM CANCELLA SCHERMO,
450 :                 REM CRSR DWN DUE VOLTE
460 :                 REM CRSR DESTRA TRE "
470 :
480 PRINT"▣▣▣▣":REM BIANCO,CRSR SINISTRA
490 :             :REM DUE VOLTE E CRSR DWN
500 :             :REM UNA SOLA VOLTA
```

```
100 REM  I CARATTERI
110 REM  SPECIALI: COME
120 REM  VENGONO INDICATI
130 REM  SULLE RIVISTE:
140 REM    COMMODORE
150 REM  E COMMODORE
160 REM  COMPUTER CLUB.
170 :
180 PRINT"[NERO]"
190 PRINT"[BIANCO]"
200 PRINT"[ROSSO]"
210 PRINT"[AZZUR]"
220 PRINT"[VIOLA]"
230 PRINT"[VERDE]"
240 PRINT"[BLEU]"
250 PRINT"[GIALLO]"
260 PRINT"[RVS]"
270 PRINT"[RVOFF]"
280 PRINT"[ARANC]"
290 PRINT"[MARR]"
300 PRINT"[ROSA]"
310 PRINT"[GRIGIO1]"
320 PRINT"[GRIGIO2]"
330 PRINT"[VERDE2]"
340 PRINT"[CELESTE]"
350 PRINT"[GRIGIO3]"
360 PRINT"[DOWN]"
370 PRINT"[RIGHT]"
380 PRINT"[UP]"
390 PRINT"[LEFT]"
400 PRINT"[HOME]"
410 PRINT"[CLEAR]"
420 :
430 REM   ESEMPI
440 PRINT"[CLEAR][2 DOWN]
   [4 RIGHT]"
450 :
460 :
470 :
480 PRINT"[BIANCO][2 LEFT
   ][DOWN]"
```

Ora anche su disco

MS-DOS & GW-BASIC EMULATOR

TRASFORMA IL TUO 64 IN UN PC IBM

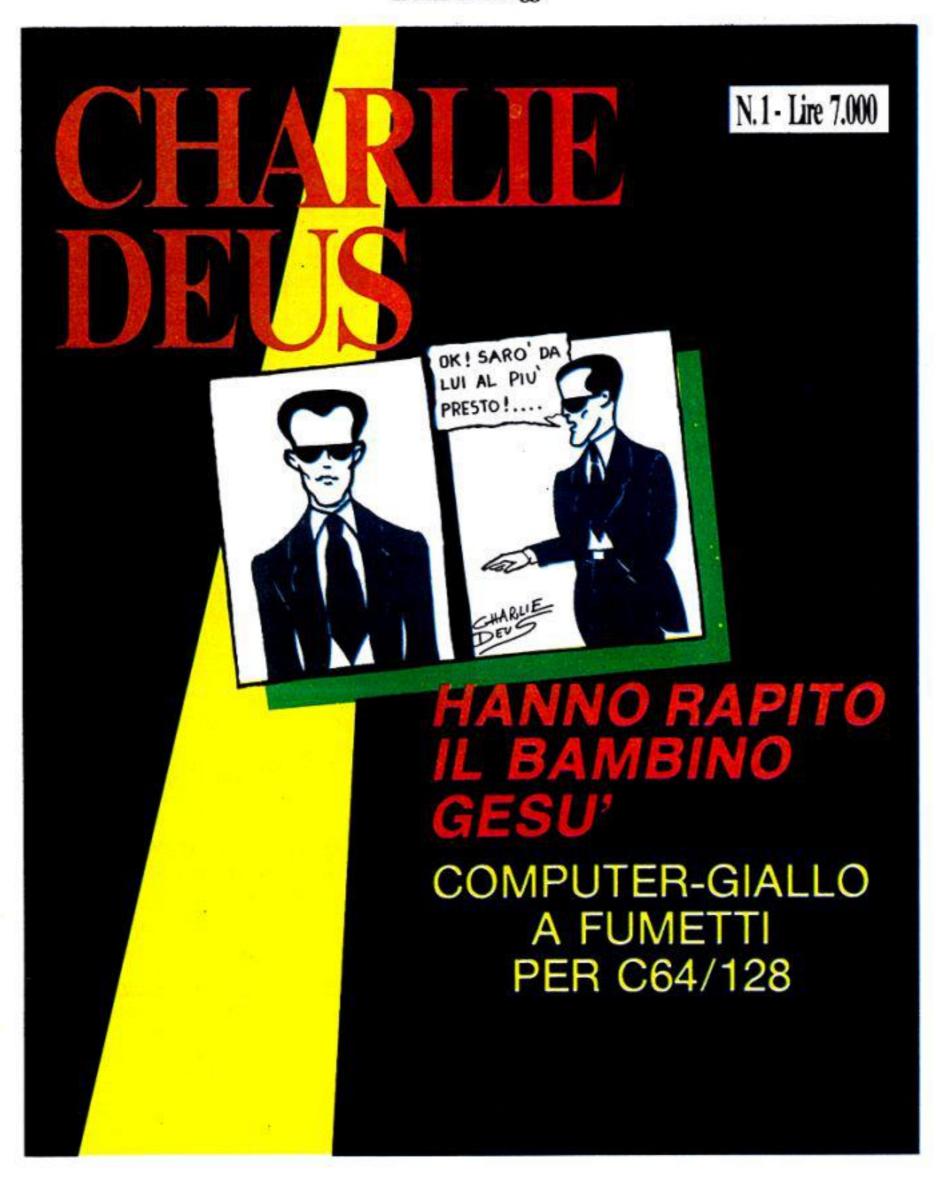
"MS-DOS & GW-BASIC emulator" è anche su disco. Per quanti hanno acquistato la versione su cassetta ed inviano la relativa prova d'acquisto, il dischetto è disponibile a lire 15.000 (+ lire 3.000 per spese di spedizione). Non occorre inviare la cassetta nè tantomeno il manualetto di istruzioni. Chi non è in possesso della cassetta può richiedere il disco ed il manualetto al prezzo normale di lire 25.000 (+ lire 3.000 per spese di spedizione).

Per una veloce evasione dell'ordine inviate un assegno bancario o circolare non trasferibile all'ordine della "Systems Editoriale" (V.le Famagosta, 75 - 20142 Milano).

Sempre un passo avanti.

Le avventure di Charlie Deus

Anteprima di un nuovo software della Systems Editoriale

di Michele Maggi

E' in edicola una nuova cassetta edita dalla Systems Editoriale intitolata "Charlie Deus" che contiene una Adventure di alto livello dal tema Natalizio.

Charlie Deus è un detective di grande fama, noto soprattutto per la sua velocità nel risolvere i casi più difficili.

Nella sua carriera di investigatore non ha mai deluso chi si è rivolto a lui e ha sempre risolto tutti gli enigmi con la flemma e il self-control tipici di un gentleman inglese.

In questa nuova serie di avventure sarai tu che, nei panni di Charlie, dovrai trovare la soluzione dei casi che ti si presenteranno.

La caratteristica che fa di questi adventure un prodotto molto particolare è la gestione del dialogo giocatore-computer che, oltre ad avere un notevole supporto grafico, presenta una particolarità unica: i dialoghi fra Charlie Deus e gli altri personaggi sono... a fumetti!

Christmas adventure

E' la vigilia di Natale e Charlie riceve una telefonata di frate Zebio, superiore dei frati di St. Joseph che, con voce concitata, gli annuncia il furto dell' Holy Child, il prezioso Bambinello che ad ogni Natale viene esposto nel presepio della cattedrale fra S. Giuseppe e la Madonna.

Il Bambinello è una scultura in legno del XII secolo con gli occhi di smeraldo e un'aureola d'oro zecchino tempestata di pietre preziose.

Questa statuina gode di grande popolarità e rappresenta una forte attrazione turistica, tanto che, nella cittadina, è sorta una piccola attività commerciale: l' Holy Child Shop, gestita da uno scultore che produce pregevoli riproduzioni del Bambinello.

Sono le ore 20 del 24 dicembre 1986 e Charlie ha solo quattro ore per ritrovare e collocare il Bambinello al suo posto in tempo per la messa di mezzanotte, sempre che riesca a trovarlo...

I comandi

Charlie è perfettamente in grado di capire quanto gli viene detto, esiste comunque una lista di verbi e comandi che permettono una più immediata comprensione:

- N = Nord
- S = Sud
- E = Est
- O = Ovest

Oltre ai comandi di movimento e ai verbi, che possono essere visualizzati in qualsiasi momento con il tasto V (+ Return), esiste una serie di comandi di una sola lettera (o simbolo) che svolgono particolari funzioni:

- I: inventario, tutto ciò di cui Charlie è in possesso verrà elencato;
- T: tempo trascorso dall'inizio dell'avventura. Ricorda, hai solo quattro ore...
- D: deduci, in particolari situazioni è possibile chiedere a Charlie di dedurre qualcosa in funzione degli elementi di cui è in possesso; una funzione di HELP molto comoda e, in certi casi, addirittura provvidenziale;
- R: ripeti, l'ultima frase apparsa nel fumetto verrà ripetuta;
- W: salvataggio dei dati;
- F: fine gioco;
- -: rallenta la velocità di scorrimento all'interno del fumetto;
- =: la riporta al valore normale;
- +: la porta ad una notevole velocità.

Tutti questi comandi devono essere seguiti dalla pressione del tasto Return.

Ogni volta che si entra in possesso di un elemento determinante per la risoluzione del caso, un piccolo stacco musicale ti avviserà che sei sulla buona strada.

HARDWARE C 64

SPEEDDOS supervelocizzatore (Montaggio a richiesta) .... DA L. 45.000
FAST LOAD il più comodo velocizzatore su cartuccia (64/128) .... L. 28.000
DOPPIO KIT GRAFICO PER MPS 802 abilita la grafica sulla 802 rendendola 803 compatibile, con uno SWITCH si potrà selezionare o il modo 802 o il modo 803, le caratteristiche della 802 non vanno perse L. 45.000
FORMEL 64 novità! nuovissimo velocizzatore per il 1541, LOAD SAVE e VERIFY ad una velocità incredibile! Contiene un potente TOOLKIT nonchè un basic esteso con moltissimi comandi utili. Viene abilitato anche un potente monitor per il linguaggio macchina. Monitor anche per il drive! .... L. 80.000
THE EXPERT superfreezer la nuova cartuccia che copia qualunque programma dalla memoria del 64 trasferendolo in **un solo file** su disco o nastro .... L. 70.000
MULTICARTRIDGE 1 (64/128) 4 favolose utility di copia e gestione disco sempre pronte all'uso perchè riunite nella stessa cartuccia (Novità ALGOBIT) .... L. 45.000
FREEZE FRAME (64/128) congela la memoria del calcolatore trasferendola su nastro o disco L. 50.000
FREEZE FRAME III ultima generazione di freezer .... L. 65.000
FLOPPY DISC DSDD 5 1/4 - 100 % Error Free (minimo 20) .... L. 2.000

Spedizione in contrassegno - Spese postali fisse L. 5.000 - Si accettano ordini telefonici

ALGOBIT s.n.c. • C.so Genova, 7 - 20123 MILANO - ✆ 02/8350804

CENTRO RIPARAZIONI RAPIDE COMMODORE

Super Data Maker

Una versione completa per trasformare in Data una qualsiasi zona di memoria

di Maurizio Dell'Abate

Salvare su nastro o disco un programma in linguaggio macchina (L.M.) non è operazione così immediata come succede con il Basic.

Tutti (o quasi) sanno che un programma in L.M. non è altro che un blocco di byte consecutivi: sarebbe sufficiente un'istruzione del tipo "BSAVE" per salvare un qualsiasi segmento di memoria, ma il C/64, manco a dirlo, non possiede un'istruzione del genere.

Siamo quindi costretti a simularla smanettando qua e là con le famigerate Poke: è sufficiente alterare i puntatori di inizio e fine Basic, effettuare un normale Save e, al momento di un futuro caricamento, utilizzare "1" come indirizzo secondario.

Il sistema più diffuso

Quella appena descritta è la tecnica più usata per salvare i programmi L.M.; esistono tuttavia altri modi per farlo. Uno dei più frequenti è quello che vedete spesso applicato nei listati L.M. di questa rivista: come avrete notato i codici vengono posti all'interno di un programma Basic, salvabile mediante un comunissimo Save.

Il codice oggetto, contenuto nelle istruzioni Data, viene letto tramite "Read" e pokato successivamente in memoria. Ciascuno dei due metodi suddetti possiede vantaggi e svantaggi. Prendiamo in esame il primo metodo ("BSAVE").

Vantaggi:
- Ogni codice L.M. occupa un solo byte in memoria.
- Al momento del caricamento da supporto magnetico, il programma viene direttamente allocato in memoria e non necessita di un ciclo di lettura e pokaggio (si evitano così stressanti attese).

Svantaggi:
- Il programma non è listabile (e come tale non potrebbe essere pubblicato su C.C.C.)
- Dopo il caricamento (da effettuare col suffisso ",1") è necessario mettere a posto i puntatori.

Prendiamo adesso in considerazione il secondo metodo (DATA).

Vantaggi:
- Il programma è listabile.
- E' un metodo comodissimo quando si utilizzano routine in linguaggio macchina da lanciare all'interno di un programma Basic.
- Non richiede l'alterazione di puntatori.
- Alla fine si ha un solo file-programma.

Svantaggi:
- Il programma impegna una cospicua quantità di memoria: ogni codice L.M. non occupa infatti un solo byte, ma ciascuna cifra di esso è memorizzata in codice ASCII. Anche i due byte del numero di linea, di link, le virgole tra un valore il successivo ed altro ancorá, contribuiscono ad aumentare ragguardevolmente la quantità di memoria occupata.
- Un ciclo deve leggere i valori dal Basic e piazzarli in una zona di memoria libera, con relativa perdita di tempo.

Quest'ultimo metodo sembra presentare gravi difetti, ma possiede anche pregi estremamente importanti.

Il programma

Nasce la necessità quindi di mettere in linee DATA un determinato segmento di memoria. Esclusa la possibilità di effettuare l'operazione "a mano" (il minimo sbaglio vi farebbe ricominciare tutto daccapo, e poi ci pensate che perdita di tempo?), abbiamo approntato un programma che esegue automaticamente il tutto.

Il programma che vi presentiamo è un Super Data Maker, ovvero un "costruttore" di linee DATA completamente automatico e dotato di alcune interessanti opzioni.

Come abbiamo già detto, un Data Maker consente la trasformazione in linee Data di una zona di memoria che rappresenta un programma in linguaggio macchina.

E' ovvio che, in precedenza, l'utilizzatore deve aver provveduto ad allocare, nella suddetta zona, il programma l.m.

Prima di utilizzare il programma di queste pagine, quindi, sarà necessario considereare che anche questo occuparà uno spazio in memoria e si dovrà provvedere a non creare pericolose sovrapposizioni.

La memoria a disposizione

La zona di memoria da porre in DATA non deve occupare la zona destinata al Basic. Potete utilizzare la zona libera che va da 679 fino a 767, il buffer del nastro (828-1023) oppure i quattro Kbyte locati da 49152 a 53247.

Nel caso aveste necessità di mettere in DATA programmi L.M. locati nella zona riservata al Basic, niente paura: trasferiteli dapprima, anche se non rilocabili, in una delle zone di memoria consentite (mediante un comunissimo ciclo For... Next). I DA-

TA, una volta creati, andranno successivamente letti e pokati, ovviamente, nella posizione originaria del programma trasferito.

Se tutto è pronto caricate Super Data Maker (SDM) e date il RUN: verranno poste alcune domande prima di "creare" i Data. Esaminiamole:

• Indirizzo di partenza: comunicate a SDM l'indirizzo del primo byte del vostro programma L.M.
• Indirizzo finale: come prima, ma l'ultimo byte.
• Tipo di notazione: può essere decimale (la consueta) oppure esadecimale; la notazione Esa è più elegante e compatta, ma necessita di una routine di conversione al momento del caricamento dei Data in memoria.
• Numero valori per linea: com'è intuibile, viene chiesta la quantità massima di valori su ogni rigo Basic; utilizzando la notazione decimale il valore massimo è 17, con l'Esa la quantità massima sale a 23 e il programma stesso, com'è intuitivo, occupa una quantità minore di memoria.
• Le successive tre domande (numero di riga iniziale, incremento e spazio dopo lo Statement DATA) sono assai semplici e scontate, ed è quindi superfluo fornirne una spiegazione.

Dopo aver risposto all'ultima domanda SDM calcola se vi sono linee Basic a sufficienza per portare a termine l'operazione: in caso negativo comunica i possibili rimedi.

Se invece tutto è andato bene non rimarrà che premere un tasto e aspettare un po' di tempo, proporzionale al numero di byte da leggere e mettere in "Data".

Appariranno, subito dopo, le righe di Data pronte per essere salvate ed essere "appese" al programma caricatore od al programma Basic che ne farà uso.

Si consiglia caldamente (soprattutto agli smemorati) di aggiungere in fondo, mediante REM, il nome della routine, la funzione che svolge, l'indirizzo di partenza (indispensabile per sapere dove pokarla!), quello di fine, le eventuali SYS di attivazione e/o di esclusione.

Come "appendere"

Date sempre una numerazione relativamente elevata alle righe di Data, rispondendo opportunamente all'apposita domanda. Questo eviterà conflitti con il programma cui verranno aggiunte.

Per effettuare un "Append" agite così:

• Spegnete e riaccendete il computer (oppure resettate).
• Caricate in memoria il programma principale, al quale andranno aggiunte le righe Data create con SDM.
• Digitate:
POKE43,PEEK(45)-2:POKE44, PEEK(46):CLR
• Se si verifica un "Illegal quantity error" (caso rarissimo), saltate ad eseguire le istruzioni poste subito dopo il presente elenco.
• Caricate, con il consueto LOAD, da nastro o disco le linee data create con SDM.
• Digitate:
POKE43,1:POKE44,8:CLR
• Se avete effettuato correttamente il tutto, chiedete LIST: i due programmi si saranno fusi in uno solo, pronto per essere salvato o utilizzato in altro modo.

Se compare il messaggio "Illegal Quantity Error", allungate di qualche byte il programma principale, utilizzando, per esempio, l'istruzione REM. Al termine dell'Append tale aggiunta potrà essere rimossa. Effettuare, quindi, le istruzioni del precedente elenco.

Un Append automatico (pubblicità)

Sul numero 15 della nostra rivista è stata pubblicata una breve routine che consente di eseguire un append in modo del tutto automatico.

Solo i possessori di unità a floppy disk potranno però usufruirne; i possessori della cassetta (che cosa aspettate a modernizzarvi?) dovranno utilizzare necessariamente la procedura "manuale" sopra descritta.

N.B.: Il programma presentato in questo numero è una versione decisamente migliorata del listato già pubblicato su C.C.C. N.19.

```
100 REM SUPER DATA MAKER - VER.
    2.0
110 :
120 REM UTILITY PER COMMODORE 6
    4/128
130 :
140 REM BY MAURIZIO DELL'ABATE
150 :
160 :
170 :
180 A=2:Y=PEEK(45)
190 IF Y<2 THEN Y=254+Y:X=1:A=0
200 POKE 253,Y-A:POKE 254,PEEK(
    46)-X
210 SYS65409:POKE 198,0
220 POKE 53281,7
230 POKE 53280,2
240 POKE 646,6
250 PRINTCHR$(147)
260 PRINT TAB(10);"SUPER DATA M
    AKER 64"
270 PRINT TAB(10);"------------
    -------"
280 PRINT:PRINT
290 PRINT TAB(9);"BY MAURIZIO D
    ELL'ABATE"
300 PRINT TAB(11);"(C) COPYRIGH
```

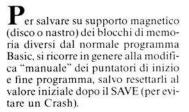
```
    T 1986"
310 PRINT:PRINT:PRINT
320 PRINT" USARE SOLO QUESTE ZO
    NE DI MEMORIA:"
330 PRINT" * 679-767...........
    ..CIRCA 90 BYTES"
340 PRINT" * 828-1023..........
    ..CIRCA 200 BYTES"
350 PRINT" * 49152-53247.......
    ..CIRCA 4 KBYTES"
360 PRINT" PER EVITARE INCONVEN
    IENTI."
370 PRINT:PRINT:PRINT
380 PRINT TAB(4);CHR$(18);" PRE
    MI UN TASTO PER CONTINUARE
    "
390 POKE 198,0:WAIT 198,1:POKE
    198,0
400 PRINTCHR$(147)
410 INPUT " INDIRIZZO DI PARTEN
    ZA";A
420 IF A<0 OR A>65535 THEN 400
430 X1=820:GOSUB 1220
440 PRINTCHR$(147)
450 INPUT " INDIRIZZO DI TERMIN
    E";B
460 IF B<0 OR B>65535 OR B<=A T
    HEN 400
470 A=B:X1=822:GOSUB 1220
480 PRINTCHR$(147)
490 PRINT" TIPO DI NOTAZIONE?"
500 PRINT:PRINT" (D).....DECIMA
    LE"
510 PRINT" (E).....ESADECIMALE"
520 POKE 198,0:WAIT 198,1:GET A
    $
530 IF A$="D" THEN POKE 2,17:GO
    TO 560
540 IF A$<>"E" THEN 520
550 POKE 2,23
560 PRINTCHR$(147):PRINT:PRINT"
     VALORE MASSIMO =";PEEK(2)
570 PRINTCHR$(19):INPUT " NUMER
    O VALORI PER LINEA";A
580 IF A<1 THEN 560
590 IF A<=PEEK(2) THEN 610
600 PRINT:PRINT:PRINT:PRINT TAB
    (11);"*** TROPPI!! ***":FOR
     Z=0 TO 999:NEXT:GOTO 560
610 POKE 826,A
620 PRINTCHR$(147)
630 PRINT" IMMETTI LA NUMERAZIO
    NE DELLA PRIMA"
640 PRINT" LINEA DI DATA. IL VA
    LORE DEVE ESSERE"
650 PRINT" MAGGIORE DI 2000 E M
    INORE DI 63000."
660 PRINT:PRINT:PRINT TAB(9);"2
    0000"
670 PRINTCHR$(145);CHR$(145):IN
    PUT " VALORE";A
680 IF A<2000 OR A>63000 THEN 6
    20
690 X1=824:GOSUB 1220
700 PRINTCHR$(147)
710 PRINT TAB(23);"10":PRINTCHR
    $(145);CHR$(145):INPUT " IN
    CREMENTO (MAX 255)";A
720 IF A<1 OR A>255 THEN 700
730 POKE 251,A
740 PRINTCHR$(147)
750 PRINT" DESIDERI UNO SPAZIO
    DI SEPARAZIONE"
760 PRINT" TRA L'ISTRUZIONE DAT
    A ED I SUCCESSIVI"
770 PRINT" VALORI? (S/N)"
780 POKE 198,0:WAIT 198,1:GET A
    $
790 IF A$="S" THEN ZX=32:GOTO 8
    20
800 IF A$<>"N" THEN 780
810 ZX=0
820 POKE 252,ZX
830 X1=820:GOSUB 1280:Z=A
840 X1=822:GOSUB 1280:Z=A-Z+1
850 PRINTCHR$(147)
860 PRINT" N. BYTES:";Z:PRINT
870 X1=824:GOSUB 1280:X=A
880 PRINT" PRIMA LINEA DI DATA:
    ";X
890 C1=Z/PEEK(826):C2=C1*PEEK(2
    51)+X
900 IF C2>63900 THEN 1430
910 PRINT:PRINT:PRINT TAB(10);C
    HR$(18);" OK, PREMI UN TAST
    O "
920 POKE 198,0:WAIT 198,1:POKE
    198,0
930 PRINTCHR$(147):PRINT TAB(1)
    ;CHR$(18);" OPERAZIONE IN C
```

```
     ORSO ":PRINT
 940 PRINT" *** ATTENDERE PREGO
     ***":PRINT:PRINT" NON PREME
     RE RUN/STOP."
 950 PRINT:PRINT:PRINT:PRINT" [
     ] ";CHR$(18);" SUPER DATA M
     AKER 64 "
 960 POKE 648,148:PRINTCHR$(147)
     :POKE 827,0
 970 X1=824:GOSUB 1280:DO=A
 980 PRINTA;CHR$(157);"DATA";CHR
     $(PEEK(252));
 990 X1=820:GOSUB 1280:G=A
1000 X1=822:GOSUB 1280:CT=0
1010 IF G=A THEN POKE 827,1:GOTO
      1070
1020 IF CT=PEEK(826)-1 THEN 1070
1030 J=PEEK(G):IF PEEK(2)=23 THE
     N GOSUB 1330:GOTO 1060
1040 X1$=STR$(J):LE=LEN(X1$)-1:X
     1$=RIGHT$(X1$,LE):IF LE=3 T
     HEN 1060
1050 Z$="":FOR CC=0 TO 2-LE:Z$=Z
     $+CHR$(48):NEXT:X1$=Z$+X1$
1060 PRINTX1$;",";:CT=CT+1:G=G+1
     :GOTO 1010
1070 J=PEEK(G):IF PEEK(2)=23 THE
     N GOSUB 1330:GOTO 1100
1080 X1$=STR$(J):LE=LEN(X1$)-1:X
     1$=RIGHT$(X1$,LE):IF LE=3 T
     HEN 1100
1090 Z$="":FOR CC=0 TO 2-LE:Z$=Z
     $+CHR$(48):NEXT:X1$=Z$+X1$
1100 PRINTX1$:G=G+1:A=G:X1=820:G
     OSUB 1220:DO=DO+PEEK(251):A
     =DO:X1=824:GOSUB 1220
1110 IF PEEK(1386)=32 THEN POKE
     1386,81:GOTO 1130
1120 POKE 1386,32
1130 PRINT"GOTO1170";
1140 POKE 631,145:POKE 632,145:P
     OKE 633,145
1150 POKE 634,13:POKE 635,13:POK
     E 198,5
1160 END
1170 IF PEEK(827) THEN 1560
1180 GOTO 970
1190 :
1200 :
1210 :
1220 HB=INT(A/256):LB=A-256*HB
1230 POKE X1,LB:POKE X1+1,HB
1240 RETURN
1250 :
1260 :
1270 :
1280 A=PEEK(X1)+PEEK(X1+1)*256
1290 RETURN
1300 :
1310 :
1320 :
1330 X1$="":REM  SENZA SPAZIO TR
     A APICI
1340 N1=INT(J/16):GOSUB 1380
1350 X1$=X1$+A$
1360 N1=J-16*N1:GOSUB 1380
1370 X1$=X1$+A$:RETURN
1380 IF N1<10 THEN A$=CHR$(48+N1
     ):RETURN
1390 A$=CHR$(55+N1):RETURN
1400 :
1410 :
1420 :
1430 PRINT:PRINT:PRINT
1440 PRINT" NON CI SONO LINEE BA
     SIC SUFFICIENTI."
1450 PRINT" ECCO COME PUOI RISOL
     VERE IL PROBLEMA:":PRINT
1460 PRINT" - DIMINUENDO LA PRIM
     A LINEA DI DATA;"
1470 PRINT" - DIMINUENDO L'INCRE
     MENTO;"
1480 PRINT" - AUMENTANDO IL N. D
     I VALORI/LINEA;"
1490 PRINT" - USANDO L'ESADEC. (
     E' PIU' COMPATTO)."
1500 PRINT:PRINT
1510 PRINT TAB(3);CHR$(18);" PRE
     MI UN TASTO PER RICOMINCIAR
     E "
1520 POKE 198,0:WAIT 198,1: RUN
1530 :
1540 :
1550 :
1560 POKE 648,4:SYS65409
1570 POKE 43,PEEK(253):POKE 44,P
     EEK(254)
1580 LIST
1590 :
1600 REM  FINE PROGRAMMA
```

Modifica del comando Save

Come aggiungere parametri "extra" al vitale comando Basic

di Roberto Morassi

Per salvare su supporto magnetico (disco o nastro) dei blocchi di memoria diversi dal normale programma Basic, si ricorre in genere alla modifica "manuale" dei puntatori di inizio e fine programma, salvo resettarli al valore iniziale dopo il SAVE (per evitare un Crash).

Un metodo meno macchinoso è quello di usare il comando "S" che si trova normalmente nei monitor in L.M. e che, comunque, richiede il preventivo caricamento del monitor stesso.

Qui proponiamo un metodo più semplice ed immediato: una breve routine in L.M. chiamata SAWEM, che modifica la sintassi del comando SAVE tramite l'inserimento di un "Wedge" o intercettatore.

Per comprenderne il funzionamento, vediamo in che modo il Sistema Operativo del computer interpreta ed esegue un comando Save appena premiamo il tasto Return.

Come ragiona il C/64

Per prima cosa vengono acquisiti dal testo, e conservati in apposite zone di memoria, i vari parametri: nome del file, numero di periferica, eventuale indirizzo secondario. Poi vengono copiati il puntatore di inizio programma Basic (43-44) e quello di fine programma (45-46) nei registri 193-194 e 174-175, rispettivamente: questi verranno usati dal Save come puntatori di "servizio".

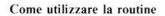
Infine, si passa ad eseguire il salvataggio vero e proprio, vale a dire la lettura dei singoli byte compresi fra i due puntatori suddetti e il loro invio alla periferica.

Quest'ultima fase, come molte altre di input/output (I/O), è "vettorizzata": prima di eseguirla, ne viene letta la locazione di inizio in un "vettore" situato nei registri 818-819 e che normalmente punta a $F5ED (decimale 62957). Essendo in RAM, questo vettore può essere modificato in modo che punti ad una nuova routine creata da noi (Wedge).

Come funziona la routine pubblicata

La routine SAWEM preleva dal testo il valore decimale del nuovo registro di inizio-Save, copiandolo (nella forma low byte-high byte) nei registri 20-21, e infine lo copia da questi ultimi nei registri 193-194 sostituendolo al valore di inizio programma Basic che vi era memorizzato.

Con procedimento analogo viene prelevato il valore del nuovo registro di fine-Save e copiato, dopo averlo incrementato di 1, nei registri 174-175. Infine si salta ad eseguire il Save del nuovo blocco di memoria, previa visualizzazione di un messaggio di controllo.

Come utilizzare la routine

Digitate il programma, salvatelo, poi date il RUN seguito da SYS 680 che ha il compito di attivare il Wedge.

La nuova sintassi è:

SAVE"NOME",P,1,INIZIO,FINE

in cui "P" è il numero della periferica (per semplicità ci riferiremo d'ora in poi all'unità a dischi, cioè: 8) e "inizio" e "fine" sono rispettivamente i numeri del primo e dell'ultimo registro del blocco di memoria da salvare.

Per comodità, salveremo la parte in L.M. dello stesso SAWEM digitando:

SAVE"SAWEM/M6",8,1,680,758

Il display della scritta "SAWEM" in giallo confermerà che la routine funziona regolarmente. In seguito potrete richiamare la routine stessa con un semplice:

LOAD"SAWEM/M6",8,1

ricordando di digitare, subito dopo, il consueto NEW per rimettere a posto i puntatori.

Finché avete il programma Basic in memoria conviene rilocare SAWEM in almeno altre due zone di memoria: potrete così utilizzare di volta in volta la versione che non si sovrappone al blocco di memoria che intendete salvare.

La rilocabilità

La routine è interamente rilocabile mediante la modifica di un solo parametro. Indichiamo, qui di seguito, alcuni esempi di rilocazioni:

Per rilocare SAWEM a partire dalla locazione 820 ($0334), sostituite "820" al posto del "680" alla riga 290. Dopo il RUN date SYS820 e, se volete:

Save"SAWEM/M8",8,1,820,898

(si attiva, ovviamente, con SYS820)

Per rilocare SAWEM a partire dalla locazione 49152 ($C000): sostituite "49152" come locazione di partenza alla riga 290. Dopo il RUN date SYS49152; poi:

SAVE "SAWEM/MC",8,1,49152, 49230

(si attiva con SYS49152).

Per disattivare il wedge, userete in ogni caso la combinazione di tasti Run/Stop e Restore o il comando SYS65418: entrambe le operazioni ripristinano tutti i vettori di I/O al valore iniziale.

I blocchi di memoria salvati con SAWEM come file-programma possono tornare utili in molti casi. Ne citiamo due:

(1) una schermata completa (in bassa risoluzione) si può salvare con:

SAVE "SCHERMO",8,1,1024,2023

e ricaricare in memoria con LOAD "SCHERMO",8,1. Naturalmente vengono salvati solo i caratteri e non i colori che appaiono sullo schermo: al momento del LOAD, i singoli caratteri assumeranno il colore che è memorizzato in quel momento nella corrispondente locazione della matrice-colore. Per una schermata di colore uniforme, è opportuno cancellare lo schermo prima di dare il LOAD.

Furbescamente si potrà salvare anche la zona contenente i mille byte della mappa colore presenti al momento della registrazione della schermata. Al caricamento, come è intuitivo, si dovranno caricare, in rapida successione, i due file (schermo + colore).

(2) una o più routine in L.M., o blocchi di dati-sprite, possono essere richiamati direttamente in memoria all'inizio di un programma, anzichè letti da una serie di DATA Basic.

Ecco come utilizzare questo metodo per le famose routine grafiche di D. Toma (CCC n. 14):

- caricate in memoria SAWEM/M6 (LOAD "SAWEM/M6",8,1)
- date il NEW
- caricate le routine grafiche (in Basic) e date il RUN. Quando la lettura dei DATA è completata, digitate: POKE 49890,84: POKE 49891,79: POKE 49892,77

Questi sono i codici corrispondenti ai caratteri T-O-M, e serviranno come "impronta digitale" per riconoscere se le routine grafiche sono già in memoria oppure no.

- attivate SAWEM con SYS 680, poi salvate il blocco di memoria con

SAVE"GRAF/M",8,1,49152,51163

A questo punto non sarà più necessario che le routine grafiche vengano caricate e attivate prima dei relativi programmi che ne fanno uso: basterà inserire in questi ultimi le seguenti righe iniziali:

```
10 IF PEEK(49890)=84 AND PEEK
(49891)=79 AND PEEK(49892)=77
THEN 30
20 LOAD "GRAF/M",8,1
30 SYS 51000: SYS 49274
40 ..............
```

Con il metodo descritto le routine grafiche verranno richiamate da disco (e attivate) una sola volta, dal primo programma che le utilizza: eventuali programmi successivi partiranno senza ripetere il caricamento.

Il programma di queste pagine (col nome: Esempio) è un altro trucco - suggerimento per caricare, sotto programma, più schermate precedentemente registrate. Dovrebbe esser noto, infatti, che quando, da programma, si carica un secondo programma, l'esecuzione riparte dalla prima riga Basic.

Ecco spiegata, quindi, la presenza di IF...THEN relativa all'esame di una locazione di memoria (e non di una variabile numerica) che, opportunamente manipolata, consente di saltare, o meno, a diversi punti del programma che, a sua volta, richiama altri programmi.

L'altra forma sintattica

La seconda routine che proponiamo (SAWEP) permette, tramite un "wedge" analogo al precedente, di salvare parti di un programma Basic (come, ad esempio, gruppi di DATA, subroutine eccetera) come se fossero programmi indipendenti, che potrete eventualmente agganciare o incorporare ad altri programmi con i vari APPEND o MERGE.

Dopo aver messo SAWEP in memoria, a partire dalla locazione 820 (anche SAWEP, comunque, è rilocabile senza modifiche), caricate o digitate il programma da cui volete estrarre i sottoprogrammi.

Attivate SAWEP con SYS820, ed usate la sintassi:

SAVE"SUBFILE",8,1,RIGA1,RIGA2

per salvare col nome SUBFILE la porzione di programma dalla riga 1 (compresa) alla riga 2 (esclusa). Esempio:

SAVE"SUBFILE",8,1,150,289

Se volete salvare la parte terminale del programma, dovrete inserire una riga provvisoria con numero più alto, da usare come "riga 2". Se i numeri di riga che indicate non si trovano nel programma, verranno usati automaticamente (come per il LIST) quelli delle righe immediatamente successive.

SAWEP utilizza la routine $A613 dell'interprete per cercare le locazioni di memoria corrispondenti ai numeri di linea indicati.

Il SUBFILE da salvare deve essere però "chiuso" correttamente, inserendo degli "zero" al posto dei due byte di "link" della riga 2: per evitare che il programma principale rimanga troncato a quel punto, e quindi inutilizzabile, SAWEP inserisce pertanto, in modo automatico, anche un secondo "wedge" nella routine di warm-start, che provvede, una volta ultimato il Save, a ripristinare il "link" della riga 2.

```
10 REM SAWEP
30 REM BY ROBERTO MORASSI
40 REM PISTOIA
130 REM ******************
140 REM SYS 820
150 REM ABILITA WEDGE
160 REM SAVE"FILE",8,1,LINE1,LI
    NE2
170 REM ******************
190 DATA  165,20,24,105,16,141,
    50,3,165
200 DATA  21,105,0,141,51,3,96,
    173,50
210 DATA  3,24,105,117,141,0,3,
    173,51,3
220 DATA  105,0,141,1,3,32,253,
    174,32
230 DATA  107,169,32,19,166,165
    ,95,133
240 DATA  193,165,96,133,194,32
    ,253,174
250 DATA  32,107,169,32,19,166,
    160,0,177
260 DATA  95,133,251,165,95,133
    ,253,165
270 DATA  96,133,254,152,145,95
    ,230,95
280 DATA  208,2,230,96,177,95,1
    33,252
290 DATA  152,145,95,230,95,208
    ,2,230
300 DATA  96,165,95,133,174,165
    ,96,133
310 DATA  175,173,50,3,24,105,1
    10,133,34
320 DATA  173,51,3,105,0,133,35
    ,169,7
330 DATA  32,36,171,76,237,245,
    158,83,65
340 DATA  87,69,80,154,32,83,22
    8,32,138,255
350 DATA  160,0,165,251,145,253
    ,200,165
360 DATA  252,145,253,76,134,22
    7
370 A=0:FOR X=0 TO 152:READ Y:P
    OKE 820+X,Y:A=A+Y:NEXT
380 IF A<>17745 THEN PRINT"ERRO
    RE NEI DATA"
390 END
```

```
10 REM SAWEM
20 :
30 REM BY ROBERTO MORASSI
40 REM PISTOIA
50 :
120 :
130 REM *************
140 REM SYS 680
150 REM ABILITA WEDGE
160 REM SAVE"FILE",8,1,START,EN
    D
170 REM *************
180 :
190 DATA  165,20,24,105,16,141,
    50,3,165
200 DATA  21,105,0,141,51,3,96,
    32,253
210 DATA  174,32,107,169,165,20
    ,164,21
220 DATA  133,193,132,194,32,25
    3,174,32
230 DATA  107,169,166,20,164,21
    ,232,208
240 DATA  1,200,134,174,132,175
    ,173,50
250 DATA  3,24,105,55,133,34,17
    3,51,3
260 DATA  105,0,133,35,169,7,32
    ,36,171
270 DATA  76,237,245,158,83,65,
    87,69,77
280 DATA  154,0
290 A=0:FOR X=0 TO 78:READ Y:PO
    KE 680+X,Y:A=A+Y:NEXT
300 IF A<>8032 THEN PRINT"ERROR
    E NEI DATA"
310 END
320 :
```

```
100 REM ESEMPIO D'USO
110 REM PER CARICARE
120 REM PIU' SCHERMATE
130 REM IN SUCCESSIONE
140 :
150 IF PEEK(679) THEN POKE 679,
    0
160 INPUT "QUALE FILE-SCHERMATA
    ";QF$
170 POKE 679,1:LOAD QF$,8,1
```

Il grande software made-in-Italy

LA VOCE III

Fa parlare e cantare il C64 secondo come lo programmi senza l'uso di campionatori nè sintetizzatori.
Tutte le parole o le canzoni così prodotte possono essere inserite come stringhe in altri programmi.

Lire 12.000

RAFFAELLO

Un programma per disegnare col tuo Comodore 64 col solo joystick senza Koala nè tavoletta grafica.
Tutti i disegni prodotti possono essere memorizzati ed uti lizzati in altri programmi.

Lire 10.000

OROSCOPO

Fa in maniera scientifica l'oroscopo personale. Il più completo programma astrologico per Commodore 64.

Lire 12.000

COMPUTER MUSIC

Un music-editor avanzato più per un programma juke-box con 27 motivi celebri di musica classica e leggera da Arcadia a Bach, Vivaldi, Zeppelin...

Lire 12.000

GESTIONE FAMILIARE

Tre programmi su cassetta che giustificano l'aggettivo "domestico" del tuo computer:
- bilancio familiare;
- dieta equilibrata;
- scheda medica familiare.

Gira su C/64/128

Lire 12.000

BANCA DATI

Un potente data base per C/64 e Spectrum disponibile anche su disco. L'edizione su cassetta contiene da un lato la versione C64 e dall'altro la versione Spectrum.

Lire 12.000

DICHIARAZIONE DEI REDDITI (740/S)

Programma aggiornato al 1986 per la dichiarazione dei redditi, modello semplificato. Per C64.

Disco: **Lire 20.000**
Cassetta: **Lire 12.000**

MATEMATICA FINANZIARIA

Pubblicato a puntate su Commodore (n.ri 13, 14 e 15) e su Personal Computer (n.ri 1, 2, 3 e 4) questo programma offre un vero e proprio corso completo di ragioneria su Commodore 64.
Se ne consiglia l'acquisto insieme agli arretrati delle riviste che ne illustrano l'uso ed il funzionamento.

Disco: **lire 20.000**
Cassetta: **lire 10.000**

Comodore 13, 14 e 15 e Personal Computer 1, 2, 3
Lire 21.000

ANALISI DI BILANCIO

Naturale completamento di "Matematica Finanziaria" questo programma consente di calcolare automaticamente tutti i ratio più significativi e di confrontare due bilanci dello stesso ente.
Il testo esplicativo è stato pubblicato su Personal Computer n.ri 2, 3, 4 e 5 che si consiglia di acquistare contemporaneamente.

Disco: **Lire 20.000**
Cassetta: **Lire 10.000**

Personal Computer 2, 3, e 5: **Lire 12.000**

ARREDARE

Un programma professionale per ottimizzare le soluzioni d'arredamento della vostra casa.
N.B. gira solo sotto Simon's Basic.

Disco: **Lire 20.000**
Cassetta: **Lire 10.000**

GRAPHIC EXPANDER 128

Un potente programma grafico per il c 128 in modo 128.

Lire 27.000

Sì, inviatemi i seguenti software al prezzo di listino + Lire 3.000 per spese di spedizione:

- ☐ RAFFAELLO
- ☐ LA VOCE III
- ☐ OROSCOPO
- ☐ COMPUTER MUSIC
- ☐ GESTIONE FAMILIARE
- ☐ BANCA DATI
- ☐ MATEMATICA FINANZIARIA/DISCO
- ☐ MATEMATICA FINANZIARIA/CASS.
- ☐ MATEMATICA FINANZIARIA/RIVISTE
- ☐ ANALISI DI BILANCIO/DISCO
- ☐ ANALISI DI BILANCIO/CASS.
- ☐ ANALISI DI BILANCIO/RIVISTE
- ☐ DICHIARAZIONE DEI REDDITI/DISCO
- ☐ DICHIARAZIONE DEI REDDITI/CASS.
- ☐ ARREDARE/DISCO
- ☐ ARREDARE/CASSETTA
- ☐ GRAPHIC EXPANDER/DISCO

Valore complessivo: Lire..

Su tale importo mi praticherete lo sconto del 10% in quanto abbonato a ☐ *Commodore Computer Club* ☐ *Personal Computer* ☐ *Computer* ☐ *VR Videoregistrare*. Pertanto vi invio la somma soltanto di lire..

☐ Desiderando ricevere le copie ordinate con la massima urgenza, accludo assegno bancario n.ro..
Banca.. per lire.. voi intestato.

☐ Contentandomi dei normali tempi postali ho inviato oggi stesso l'importo di lire.......................... a mezzo C/C postale N. 37952207 intestato a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Nome ....
via .... **N.ro.** .... **telefono** ....
CAP .... **Città** ....

C 64

Un mondo di cubi colorati

Simuliamo, con il computer, elementari esperienze di logica

di Sandro Certi & Franco Toldi

Un bambino gioca con i suoi cubi colorati sul parquet della sua stanza.

Possiede solo quattro cubi: uno rosso, uno giallo, uno verde e uno blu, sparsi sul pavimento; il bimbo inizia a mettere quello giallo sopra quello rosso.

L'equilibrio è precario, ma alla fine stanno in piedi. Poi impila su quello giallo anche quello blu... Sorride soddisfatto: il suo obiettivo è stato raggiunto.

Poi, si sa, il bambino è curioso e inventivo; vorrebbe vedere che effetto fa un cubo rosso sopra quello verde.

Il cubo verde è libero sul pavimento, ma quello rosso è sovrastato da quello giallo e, ancora sopra, da quello blu.

Naturalmente la tentazione è di distruggere la pila...

Il contenuto educativo della manipolazione dei cubi colorati risulta valido proprio perchè gradualmente il bambino impara una procedura sempre meno distruttiva.

Prende il cubo blu dalla pila e lo mette sul pavimento. Poi fa altrettanto con il cubo giallo. Ora il cubo rosso non è più sovrastato da altri cubi. Il bimbo lo prende e lo mette sopra il cubo verde. Obiettivo raggiunto...

In sostanza, nel suo mondo dei cubi colorati, il bambino ha introdotto una regola: prima di afferrare un cubo, per spostarlo dalla sua posizione originaria, occorre fare in modo che la sua sommità sia libera, cioè che nessun altro cubo sia appoggiato so-

pra. Se invece ciò non si verifica, allora è bene rimuovere dapprima il cubo superiore e poi afferrare quello voluto.

Semplice, vero?

Certamente semplice e molto istruttivo.

Il valore formativo del gioco dei cubi colorati consiste nel fatto che il bambino inizia ad operare fissandosi degli obiettivi. Poi, nel realizzarli, si accorge che il suo obiettivo ha delle precondizioni che devono essere a loro volta realizzate; sono questi gli stadi intermedi di un processo complesso che porterà alla realizzazione dell'obiettivo originale prefissato.

SHRDLU: un programma intelligente

Nei primi anni settanta Terry Winograd, ricercatore di A/I (Intelligenza Artificiale) al MIT (Massachusetts Institute of Technology), presentò un programma intelligente che manipolava in modo automatico un mondo immaginario di figure geometriche (cubi, piramidi, cilindri) in uno spazio delimitato secondo il metodo della pianificazione decisionale finalizzata ad un obiettivo.

Il nome del programma, forse bizzarro, era SHRDLU, una successione delle lettere dell'alfabeto inglese, dalla settima alla dodicesima, in ordine di frequenza decrescente.

L'utente del programma indica una certa disposizione finale da realizzare ed il computer, rispettando alcune semplici regole, esegue tutte le operazioni necessarie per realizzare la disposizione richiesta.

Naturalmente, come il bimbo che abbiamo osservato prima, non si limita a distruggere la disposizione di partenza e a costruire quella finale, ma muove tutti i pezzi necessari fino a trasformare la disposizione di partenza in quella richiesta.

Se, ad esempio, dovesse eseguire il desiderio del bimbo che abbiamo seguito precedentemente, cioè di mettere il cubo rosso (che è sovrastato da quello giallo e da quello blu) su quello verde, seguirebbe esattamente questa procedura:

- **sopra il cubo rosso c'è un altro cubo?**
- **si: quello giallo**
- **sopra il cubo giallo c'è qualcosa?**
- **si: quello blu**
- **sopra il cubo blu c'è qualcosa?**
- **no: allora posso prendere il cubo blu e spostarlo**
- **sopra il cubo giallo c'è qualcosa?**
- **no: allora posso prendere il cubo giallo e spostarlo**
- **sopra il cubo rosso c'è qualcosa?**
- **no: allora posso prenderlo e spostarlo**
- **dove? sopra il cubo verde**
- **sul cubo verde c'è qualcosa?**
- **no: allora posso mettervi sopra quello rosso**
- **obiettivo finale realizzato**

Come si vede la procedura è meno semplice di quanto ci si aspetta: è composta di numerose verifiche e di spostamenti successivi che l'utente, nel richiedere di mettere il cubo rosso sopra quello verde, non aveva certo indicato nei dettagli al computer.

Ricevuto il comando dall'utente, il computer esamina la situazione di partenza e determina un piano di successive azioni che permetteranno di realizzare da disposizione richiesta.

E' questo un caso tipico di Pianificazione decisionale con concatenazione a ritroso (bachward chaining).

Nel caso semplice del mondo dei cubi le azioni possibili sono soltanto due.

La prima è: liberare la sommità di un cubo. Questa azione ha due precondizioni per essere eseguibile: il cubo da liberare deve essere sovrastato da un altro cubo; questo secondo cubo deve essere libero, cioè non sovrastato da nulla.

La seconda azione è: mettere un cubo sopra un altro. Anche questa azione ha due precondizioni per essere eseguibile: il primo cubo deve essere libero per poterlo muovere; anche il secondo cubo deve essere libero per ricevere sopra di esso il primo cubo.

Ricevendo l'ordine di mettere il cubo X sopra il cubo Y il computer si predispone a eseguire l'azione corrispondente; ne verifica per prima cosa le precondizioni (sia X che Y devono essere liberi) se sono realizzate, esegue l'azione "metto il cubo X su quello Y"; altrimenti, attraverso azioni di tipo "libera", libererà la sommità dei cubi X e Y (o quello che non era libero) finchè le precondizioni per l'azione "metti" non siano verificate. Solo a questo punto il computer esegue l'azione che permetterà di realizzare la disposizione richiesta dall'utente.

Il mondo dei cubi colorati

Il programma che presentiamo si fonda sui concetti che abbiamo esposto.

Consiste di semplici routine il cui richiamo appropriato viene governato dalla routine principale (linee 100-240) detta "supervisore".

Prima di analizzare in dettaglio le singole routine anticipiamo qualche informazione utile per il lettore.

Lanciato il programma, compaiono immediatamente le istruzioni che comunque potranno essere richiamate in ogni momento immettendo in input la stringa ISTRUZIONI.

Il mondo dei cubi colorati manipola, su una traccia di piano comprendente sei posizioni orizzontali, la disposizione di quattro cubi colorati secondo i comandi ricevuti in input dall'utente.

I comandi disponibili sono:

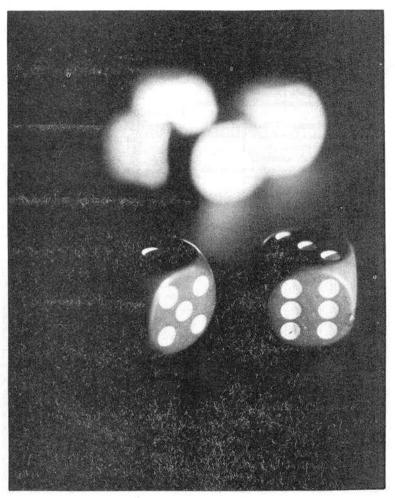
GENERA

questo comando genera, appunto, una disposizione casuale dei quattro cubi lungo le sei posizioni della traccia di piano. All'inizio, ovviamente, ogni cubo occupa una posizione e nessun cubo è sovrastato da un altro cubo.

LIBERA IL CUBO XXX

(esempio: libera il cubo verde); il comando, da usarsi con questo formato esatto (altrimenti il programma segnala COMANDO SCORRETTO e richiede un nuovo input), libera il cubo XXX da ogni eventuale cubo che sia appoggiato alla sua sommità, spostando il cubo sovrastante in una posizione libera della traccia di piano.

METTI IL CUBO XXX SU QUELLO YYY

(esempio: metti il cubo rosso su quello giallo); anche questo comando, come tutti gli altri, deve essere usato nel preciso formato indicato. A partire

dalla disposizione attuale che lo schermo visualizza, questo comando sposta il cubo XXX e lo mette al di sopra di quello YYY, naturalmente verificando o realizzando le precondizioni necessarie all'azione di tipo "metti".

Tutti i comandi vengono eseguiti dopo che il programma ha esposto esplicitamente sullo schermo il piano di azione che ha scelto per realizzare il comando stesso. A questo punto sollecita l'utente con BATTI UN TASTO. Ottenuto il via, il programma evidenzierà la nuova disposizione ottenuta e si predisporrà per un nuovo comando.

Suggeriamo all'utente di provare con pazienza molte operazioni, da diverse disposizioni di partenza, per apprezzare bene le capacità del programma anche nei casi limite.

Il mondo dei cubi colorati, a dispetto della sua brevità, è un programma che ha una discreta funzione didattica per sperimentare la realizzazione di programmi (più) intelligenti...

Le routine

Seguiamo ora il listato del programma.

La routine supervisore (linee 100-240) richiamando le oppurtune routine, inizializza il programma; genera e visualizza la prima disposizione dei cubi colorati; riceve in input il comando dell'utente e lo analizza: identificando con P$(1) la prima parola di cui è composta la frase-comando, richiama la routine del comando richiesto: lo esegue, visualizza la disposizione ottenuta e si predispone ad un nuovo comando.

La routine inizializzazione (250-330), che sarà eseguita una sola volta al lancio del programma, dimensiona una matrice M$ con indici 5 e 6. Sei, infatti, sono le posizioni possibili sulla traccia di piano in çui saranno disposti i cubi; mentre per poter manipolare anche le situazioni in cui tutti i quattro cubi sono sovrapposti in una unica pila occorrono non quattro ma cinque indici che rappresenteranno le posizioni verticali.

La matrice M$ costituisce la rappresentazione che il computer ha della disposizione dei cubi colorati in ogni momento e in essa avverranno gli spostamenti di manipolazione dei cubi.

Nella routine vengono anche determinate le stringhe rappresentative dei quattro cubi colorati (sono semplicemente dei quadrati colorati) e di una stringa V$ che rappresenta una posizione non occupata.

La formalizzazione di V$, in analogia con le stringhe rappresentative dei cubi colorati, è stata scelta, per comodità di visualizzazione, come un quadrato colorato in bianco cioè del colore del fondo dello schermo.

La routine "Genera" (340-500) riempie dapprima tutta la matrice con la stringa rappresentativa della posizione non occupata, cioè V$; successivamente dispone in modo casuale le quattro stringhe rappresentative dei quattro cubi colorati lungo le posizioni inferiori della matrice, verificando che ogni stringa rappresentativa di un cubo colorato sia collocata in una posizione non occupata.

La routine di visualizzazione (510-600) scrive semplicemente sullo schermo le stringhe di cui è composta la matrice.

Viene anche visualizzata la traccia

di piano su cui sono appoggiati i cubi e numerate le relative posizioni.

La routine di input (610-690) compie due operazioni: riceve l'input dell'utente contenente la frase-comando e la scinde nelle singole parole di cui è costituita. Il ciclo (650-680) scorre, carattere per carattere, l'intera stringa di input alla ricerca del carattere SPAZIO, che separa tra di loro le singole parole. Le sottostringhe così ottenute vengono immesse in un vettore P$(P) in cui l'indice P indica l'ordine con cui le parole compaiono nella frase-comando. Da ora in poi P$(1) sarà dunque una sottostringa della stringa input, capace di individuare il tipo di comando richiesto dall'utente (METTI, LIBERA, GENERA, oppure ISTRUZIONI).

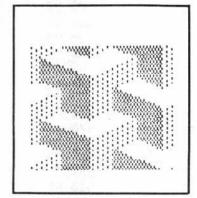
Infatti il supervisore a questo punto, controllando P$(1) richiama la routine capace di eseguire il comando richiesto dall'utente (linee 170-210).

Il programma utilizza due routine di servizio (linee 1130-1190 e 1400-1430). La prima serve a individuare per ogni stringa rappresentativa del disegno del quadrato colorato la corrispondente stringa in caratteri scritti: se in matrice è conservata C1$, allora nella comunicazione scritta con l'utente il programma utilizzerà la stringa ROSSO; e così via.

La seconda routine di servizio controlla, attraverso il valore che attribuisce alla variabile OK, che i comandi dell'utente riguardino soltanto i colori dei cubi previsti dal programma. Se l'utente pretendesse di manipolare un cubo ad esempio MARRONE, allora la variabile OK resterebbe al valore zero e comparirebbe il segnale COMANDO SCORRETTO.

Veniamo ora alle due routine più importanti, quelle realizzative dei comandi di tipo LIBERA e di tipo METTI.

La routine LIBERA (700-980) per prima cosa individua il colore del cubo che deve essere liberato. Il colore sarà rappresentato dalla quarta parola di cui è costituita la frase-comando (LIBERA IL CUBO XXX) e conservato in P$(4).

Ad ogni valore di P$(4) corrisponderà nella matrice M$ la stringa rappresentativa che viene individuata nelle linee 730-760.

Le linee 770-820 scorrono il contenuto della matrice M$ allo scopo di individuare la posizione in cui è conservata la stringa rappresentativa del cubo oggetto del comando. Non vengono usate le istruzioni FOR...NEXT per non rischiare di lasciare aperto pericolosamente il ciclo. Una volta individuata la posizione cercata, i valori indice vengono conservati nelle variabili R e S.

A questo punto il programma esamina le precondizioni per poter eseguire l'azione di tipo LIBERA, cioè se il cubo in oggetto sia o meno sovrastato da un altro cubo. Viene dunque valutata la posizione della matrice M$ caratterizzata dagli indici R+1 e S.

Se in posizione (R+1,S) nella matrice si trova la stringa V$ rappresentativa di una posizione non accupata allora in programma comunica NON C'E' NULLA SUL CUBO XXX e restituisce il controllo al supervisore. Se invece in (R+1,S) si trova una stringa rappresentativa di un cubo colorato, allora occorrerà muoverlo in una posizione libera della traccia di piano e immettere al suo posto nella matrice la stringa V$, rappresentativa di una posizione non occupata (linee 900-960).

La condizione per muoverlo è naturalmente che quest'ultimo cubo sia a sua volta libero, dunque la routine funziona in semplice modo ricorsivo governato attraverso il valore della variabile K nelle linee 870-880.

La routine LIBERA opera però in due diverse condizioni. O per eseguire direttamente un comando di tipo LIBERA fornito dall'utente; oppure come precondizione per una azione di tipo METTI.

Abbiamo infatti visto che per eseguire il comando METTI IL CUBO XXX SU QUELLO YYY occorre che il cubo XXX venga liberato per poterlo prendere e spostare sopra il cubo YYY, che a sua volta deve essere libero.

La variabile F indica alla routine LIBERA se essa viene richiamata direttamente dal supervisore o dalla routine METTI. Ora sarà semplice vedere la funzione della linea 850 che viene eseguita soltanto quando si tratta di liberare un cubo come precondizione per metterlo su di un altro.

La routine METTI (linee 990-1120) appare sicuramente di più facile comprensione.

Il cubo XXX del comando di tipo METTI sarà rappresentato dalla stringa P$(4) mentre quello YYY da P$(7). Entrambe le stringhe vengono controllate attraverso la routine di servizio (GOSUB 1400).

La routine METTI individua il cubo XXX e, attraverso il richiamo della routine LIBERA, verifica o realizza le precondizioni per muoverlo; successivamente individua il cubo YYY e, sempre attraverso la routine LIBERA, verifica o realizza le precondizioni perchè possa essere sormontato dal cubo XXX. La prima volta la routine LIBERA viene richiamata a partire dalla linea 730, la seconda a partire dalla linea 720.

A questo punto nella matrice M$ vengono realizzati gli opportuni spostamenti di posizione delle stringhe rappresentative dei cubo colorati e delle posizioni che diventano non occupate; poi il controllo viene restituito al supervisore che visualizzerà la nuova disposizione del mondo dei cubo colorati.

```
100 REM  IL MONDO DEI CUBI
110 REM  SUPERVISORE
120 GOSUB 250:REM  INIZIALIZZA
130 GOSUB 1200:REM  ISTRUZIONI
140 GOSUB 340:REM  GENERA
150 GOSUB 510:REM  VISUALIZZA
160 GOSUB 610:REM  INPUT E RICO
    NOSCE COMANDI
170 IF P$(1)="GENERA" THEN GOSU
    B 340
180 IF P$(1)="LIBERA" THEN GOSU
    B 700
190 IF P$(1)="METTI" THEN GOSUB
     990
200 IF P$(1)="ISTRUZIONI" THEN
    GOSUB 1200
210 IF P$(1)="" THEN END
220 PRINT"BATTI UN TASTO"
230 GET X$:IF X$="" THEN 230
240 GOTO 150
250 REM  INIZIALIZZA
260 POKE 53281,1:POKE 53280,1:P
    RINT"[CLEAR][BLEU]"
270 DIM M$(5,6)
280 C1$="[ROSSO][RVS]┌┐[DOWN][2
     LEFT]└┘[RVOFF][BLEU]":REM
     ROSSO
290 C2$="[GIALLO][RVS]┌┐[DOWN][
     2 LEFT]└┘[RVOFF][BLEU]":REM
      GIALLO
300 C3$="[VERDE][RVS]┌┐[DOWN][2
     LEFT]└┘[RVOFF][BLEU]":REM
     VERDE
310 C4$="[BLEU][RVS]┌┐[DOWN][2
    LEFT]└┘[RVOFF][BLEU]":REM
    BLU
320 V$="[BIANCO][RVS]  [DOWN][2
     LEFT]  [RVS][BLEU]":REM  V
    UOTO
330 RETURN
340 REM  GENERA UNA SITUAZIONE
    NUOVA
350 FOR X=1 TO 5
360 FOR Y=1 TO 6
370 M$(X,Y)=V$
380 NEXT:NEXT
390 Y1=INT(RND(1)*6)+1
400 Y2=INT(RND(1)*6)+1
410 IF Y2=Y1 THEN 400
420 Y3=INT(RND(1)*6)+1
430 IF Y3=Y2 OR Y3=Y1 THEN 420
440 Y4=INT(RND(1)*6)+1
450 IF Y4=Y3 OR Y4=Y2 OR Y4=Y1
    THEN 440
460 M$(1,Y1)=C1$
470 M$(1,Y2)=C2$
480 M$(1,Y3)=C3$
490 M$(1,Y4)=C4$
500 RETURN
510 REM  VISUALIZZA IL MONDO DE
    I CUBI
520 PRINT"[CLEAR]"
530 FOR X=5 TO 1 STEP -1
540 PRINT TAB(8);
550 FOR Y=1 TO 6
560 PRINTM$(X,Y);"[UP][2 RIGHT]
    ";
570 NEXT:PRINT:PRINT
580 NEXT:PRINT TAB(8)"__________
    ___________________"
590 PRINT TAB(8)" 1   2   3   4
       5   6"
600 RETURN
610 REM  INPUT E INDIVIDUA COMA
    NDI
620 PRINT:I$="":KK=1:P=0:F=0
630 FOR X=1 TO 9:P$(X)="":NEXT
640 INPUT I$:I$=I$+" "
650 FOR SP=KK TO LEN(I$)
660 IF P>9 THEN 680
670 IF MID$(I$,SP,1)=" " THEN P
    =P+1:P$(P)=MID$(I$,KK,SP-KK
    ):KK=SP+1
680 NEXT:PRINT
690 RETURN
700 REM  LIBERA
710 P$=P$(4)
720 A=4:GOSUB 1400:IF OK=0 THEN
     PRINT"COMANDO SCORRETTO":G
```

```
    OTO 980
 730 IF P$="ROSSO" THEN CX$=C1$
 740 IF P$="GIALLO" THEN CX$=C2$
 750 IF P$="VERDE" THEN CX$=C3$
 760 IF P$="BLU" THEN CX$=C4$
 770 X=5
 780 Y=1
 790 IF M$(X,Y)=CX$ THEN 830
 800 IF Y<6 THEN Y=Y+1:GOTO 790
 810 IF X>1 THEN X=X-1:GOTO 780
 820 PRINT"NON TROVO IL CUBO":GO
     TO 980
 830 R=X:S=Y
 840 IF M$(R+1,S)=V$ THEN PRINT"
     NON C'E NULLA SUL CUBO ";:A
     $=M$(R,S):GOSUB 1130
 850 IF M$(R+1,S)=V$ AND F=-1 TH
     EN PRINT"LO PRENDO"
 860 IF M$(R+1,S)=V$ THEN 980
 870 IF M$(R+2,S)=V$ THEN K=1:GO
     TO 890
 880 K=3:IF M$(R+3,S)=V$ THEN K=
     2
 890 FOR J=K TO 1 STEP -1
 900 PRINT"DEVO MUOVERE IL CUBO
     ";
 910 A$=M$(R+J,S):GOSUB 1130
 920 MN=INT(RND(1)*6)+1
 930 IF M$(1,MN)<>V$ THEN 920
 940 PRINT"LO PRENDO E LO MUOVO
     IN ";MN
 950 M$(1,MN)=M$(R+J,S):M$(R+J,S
     )=V$
 960 NEXT
 970 GOTO 840
 980 RETURN
 990 REM  METTI
1000 A=4:GOSUB 1400:IF OK=0 THEN
      PRINT"COMANDO SCORRETTO":G
     OTO 1120
1010 A=7:GOSUB 1400:IF OK=0 THEN
      PRINT"COMANDO SCORRETTO":G
     OTO 1120
1020 F=-1:P$=P$(4)
1030 GOSUB 730:T=R:U=S:CC$=M$(T,
     U):M$(R,S)=V$
1040 P$=P$(7)
1050 F=0:GOSUB 720
1060 PRINT"MUOVO IL CUBO ";
1070 A$=CC$:GOSUB 1130
1080 PRINT"SU QUELLO ";
1090 A$=M$(R,S):GOSUB 1130
1100 M$(T,U)=V$
1110 M$(R+1,S)=CC$
1120 RETURN
1130 REM  COLORI SCRITTI
1140 IF A$=C1$ THEN PRINT"ROSSO"
1150 IF A$=C2$ THEN PRINT"GIALLO
     "
1160 IF A$=C3$ THEN PRINT"VERDE"
1170 IF A$=C4$ THEN PRINT"BLU"
1180 A$=""
1190 RETURN
1200 REM  ISTRUZIONI
1210 PRINT"[CLEAR]"
1220 PRINT"IL MONDO DEI CUBI"
1230 PRINT:PRINT"QUATTRO CUBI CO
     LORATI VENGONO SPOSTATI "
1240 PRINT"IN UN PIANO COSTITUIT
     O DI SEI CASELLE"
1250 PRINT"SECONDO COMANDI IMPAR
     TITI E REGOLE CHE"
1260 PRINT"IL COMPUTER VERIFICA
     AUTOMATICAMENTE."
1270 PRINT"IL PROGRAMMA POSSIEDE
      TRE ISTRUZIONI DA"
1280 PRINT"USARSI NEL PRECISO FO
     RMATO SEGUENTE:":PRINT:PRIN
     T"[RVS]ISTRUZIONI":PRINT
1290 PRINT"[RVS]GENERA"
1300 PRINT"GENERA UNA NUOVA SITU
     AZIONE DISPONENDO  A CASO I
      CUBI NEL PIANO."
1310 PRINT:PRINT"[RVS]LIBERA"
1320 PRINT"ES: LIBERA IL CUBO XX
     X"
1330 PRINT:PRINT"[RVS]METTI"
1340 PRINT"ES: METTI IL CUBO BLU
      SU QUELLO ROSSO"
1350 IF XX=-1 THEN 1390
1360 PRINT:PRINT"BATTI UN TASTO"
1370 GET X$:IF X$="" THEN 1370
1380 XX=-1
1390 RETURN
1400 REM  CONTROLLO
1410 OK=0
1420 IF P$(A)="ROSSO" OR P$(A)="
     GIALLO" OR P$(A)="VERDE" OR
      P$(A)="BLU" THEN OK=1
1430 RETURN
```

LA DIDATTICA E' SYSTEMS

VELOCISSIMO BASIC

per C64/128, C16 & Plus 4, MSX e Spectrum.
Corso completo in 13 lezioni su 4 cassette interamente gestite dal computer.
Il corso è diviso in 4 parti, ciascuna delle quali contiene la versione specifica per il computer cui si riferisce

Lire 24.000

ASSEMBLER TUTOR

Un corso completo sull'assembler del C/64 in 8 lezioni interamente gestite dal computer, più un programma *MONITOR.*

Lire 12.000

LOGO 64

La più originale versione del LOGO. Programma non protetto in basic facilmente personalizzabile.

Lire 10.000

24 ORE BASIC

Il corso di basic più veloce per C/64 13 lezioni su 4 cassette con una introduzione "parlata".
Tutto il basic senza libri nè dispense.

Lire 24.000

µ PASCAL PER C64

Il volume introduttivo sul Pascal della collana i "libri Systems" completata dalla cassetta con il programma compilatore.

Lire 19.500 (Libro + cassetta)

MS-DOS & GW-BASIC emulator

Il primo programma in grado di emulare sul C/64 il sistema operativo ed il più diffuso basic dei PC Ibm.

Lire 12.000 su cassetta
Lire 25.000 su disco

Si, inviatemi al più presto il seguente software, al prezzo contrassegnato, più lire 3.000 per spese di spedizione:

☐ VELOCISSIMO BASIC (24.000)
☐ ASSEMBLER TUTOR (12.000)
☐ MS-DOS & GW-BASIC EMULATOR
  ☐ versione cassetta (Lire 12.000)
  ☐ versione disco (Lire 25.000)
☐ 24 ORE BASIC (Lire 24.000)
☐ PASCAL PER COMMODORE 64 (Libro +
☐ cassetta lire 19.500)
☐ LOGO 64 (Lire 10.000)

Importo totale lire: ....

Su tale importo mi praticherete lo sconto del 10% in quanto abbonato a ☐ *Commodore Computer Club* ☐ *Personal Computer* ☐ *Computer* ☐ *VR Videoregistrare.* Pertanto vi invio la somma soltanto di lire..........

☐ Desiderando ricevere le copie ordinate con la massima urgenza, accludo assegno bancario n.ro.......... Banca.......... per lire.......... voi intestato.

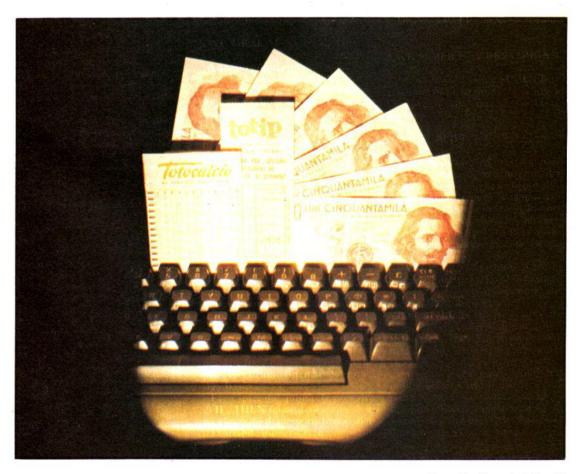
☐ Contentandomi dei normali tempi postali ho inviato oggi stesso l'importo di lire.......... a mezzo C/C postale N. 37952207 intestato a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Nome ....
via .... **N.ro.** .... **telefono** ....
CAP .... **Città** ....

Rintracciamo la colonna vincente

Se avete realizzato un buon punteggio con il programma TOT 13, non vi resta che individuare, tra quelle giocate, le schedine vincenti...

di Antonio Pastorelli

Questo programma fa seguito a TOT13, pubblicato sul numero 35 di C.C.C, che permetteva lo sviluppo di sistemi condizionati.

Ricorderete, infatti, che dopo aver impostato e sviluppato un sistema con TOT13, avete la possibilità di salvare le colonne elaborate, sia per consultarle più volte (soprattutto nella noiosa fase della trascrizione), sia per avere un archivio al quale fare ricorso per controllare i punteggi conseguiti.

Il programma di queste pagine caricherà in memoria le colonne precedentemente giocate e controllerà, colonna per colonna, i punteggi ottenuti indicando non solo quanti 13, 12, 11 (e perfino 10) avete totalizzato ma anche la schedina che contiene gli alti punteggi riscontrati.

Per avere questa ultima comodità è necessario, ovviamente, trascrivere una per una le colonne nell'ordine in cui appaiono sullo schermo al momento dell'Otput di TOT/13 e numerare le schedine prima di consegnarle alla ricevitoria.

Come usare il programma

L'uso del programma è molto semplice: dopo il RUN dovrete indicare il nome del file con cui avete memorizzato le colonne giocate. In TOT/13 il nome del file era "fisso" (cioè: TOT) ma è probabile che qualche lettore abbia modificato il programma per memorizzare più file sullo stesso disco. La possibilità di indicare il nome, inoltre, risulta indispensabile per i lettori che volessero utilizzare anche il programma "Sistemi Ridotti" pubblicato in questo stesso numero.

Pertanto la domenica sera, dopo la comucazione ufficiale della colonna vincente, digitatela quando il programma la richiede e confermate premendo il tasto "S" (oppure, in caso di errore di digitazione, non si conferma e si ripete l'introduzione).

A questo punto il computer confronterà la colonna vincente con ciascuna colonna memorizzata e sospende l'elaborazione in due casi: quando ha terminato il controllo oppure quando ha individuato un punteggio maggiore di 10. In quest'ultimo caso viene visualizzata l'informazione utile per rintracciare la schedina che contiene la colonna individuata; per proseguire basterà premere un tasto qualsiasi.

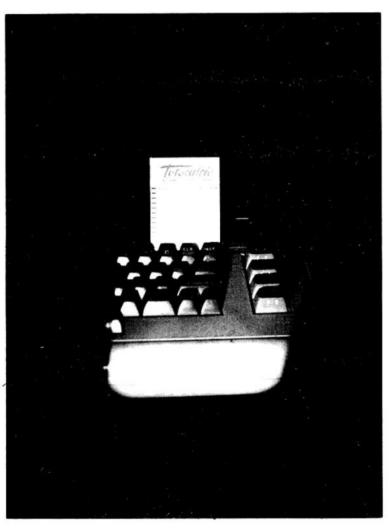
```
100 REM CALCOLO PUNTEGGI CONSEG
    UITI CON TOT/13
110 REM PER C/16 PLUS/4 E C/64
120 :
130 REM BY ANTONIO PASTORELLI
140 :
150 POKE 55,251:POKE 56,33:CLR
160 INPUT "[CLEAR]NOME DEL FILE
     REGISTRATO";W0$
170 X1$=CHR$(213):X2$=CHR$(192)
    :X3$=CHR$(178):X4$=CHR$(201
    ):X5$=CHR$(221):X6$=CHR$(21
    9)
180 X7$=CHR$(171):X8$=CHR$(179)
    :X9$=CHR$(202):XA$=CHR$(177
    ):XB$=CHR$(203)
190 GOSUB 840:REM COMPILA ROUTI
    NE L/M
200 PRINT"[CLEAR][RVS]
      P    U    N    T    I
     "
210 FOR J=1 TO 1000:NEXT
220 PRINT"[2 DOWN]PREMI UN TAST
    O E, IN SEGUITO, INTRODUCI
    LA COLONNA VINCENTE."
230 REM INTRODUZIONE COLONNA VI
    NCENTE
240 GET A$:IF A$="" THEN 240
250 PRINT"[CLEAR]"X1$X2$X2$X2$X
    3$X2$X4$:FOR X=1 TO 3:FOR J
    =1 TO 3:PRINTX5$"   "X5$" "
    X5$
260 NEXT
270 PRINTX7$X2$X2$X2$X6$X2$X8$:
    NEXT:FOR J=1 TO 4:PRINTX5$"
       "X5$" "X5$:NEXT
280 PRINTX9$X2$X2$X2$XA$X2$XB$
290 PRINT"[HOME]":X=0:FOR J=1 T
    O 13:X=X+1:IF X>3 AND J<12
    THEN X=1:PRINT"[DOWN]";
300 PRINT TAB(4-LEN(STR$(J)));J
    :NEXT
310 PRINT"[HOME]":X=0:FOR J=1 T
    O 13
320 GET A$:IF A$="1" THEN A=1:G
    OTO 360
330 IF A$="X" THEN A=2:GOTO 360
340 IF A$="2" THEN A=3:GOTO 360
350 GOTO 320
360 X=X+1:IF X>3 AND J<12 THEN
    X=1:PRINT"[DOWN]";
370 POKE 8703+J,A:PRINT TAB(5);
    "[RVS]";A$:NEXT
380 FOR J=1 TO 700:NEXT
390 PRINT"[2 DOWN][RVS]CONFERMI
     (S/N)?"
400 GET A$:IF A$="S" THEN 430
410 IF A$="N" THEN 200
```

```
420 GOTO 400
430 IF ZX THEN 510
440 PRINT"[CLEAR]LE COLONNE GIO
    CATE SONO STATE REGISTRATES
    U [RVS]D[RVOFF]ISCO O SU [R
    VS]N[RVOFF]ASTRO?"
450 PRINT"[DOWN](INSERISCI IL D
    ISCO OPPURE POSIZIONA IL NA
    STRO CHE CONTIENE I DATI)"
460 REM CARICAMENTO DATI COLONN
    E GIOCATE
470 GET A$:IF A$="D" THEN OPEN
    1,8,12,W0$+",S,R":GOTO 500
480 IF A$="N" THEN OPEN 1,1,0,W
    0$:GOTO 500
490 GOTO 470
500 INPUT#1,NT:NR=NT:PRINT"[CLE
    AR][RVS]ATTENDERE:[RVOFF] C
    ARICAMENTO DATI IN CORSO...
    ."
510 POKE 8718,INT(NT/256):POKE
    8717,NT-INT(NT/256):POKE 87
    20,0:POKE 8721,0
520 IF ZX THEN 550
530 FOR X=1 TO NR:FOR J=1 TO 13
540 INPUT#1,A:POKE 8810+J+X*13,
    A:NEXTJ,X:CLOSE 1:ZX=1
550 PRINT"[CLEAR]VERIFICA PUNTE
    GGI CONSEGUITI....":FOR J=1
     TO 300:NEXT
560 SYS8724
570 REM LETTURA RISULTATI
580 NR=PEEK(8721)*256+PEEK(8720
    )
590 B=NR/4:IF B-INT(B)=0 THEN S
    C=B:CO=4:GOTO 610
600 CO=(B-INT(B))*4:SC=INT(B)+1
610 A=PEEK(8719):IF NR=NT THEN
    FI=1
620 IF A<10 THEN 690
630 IF A=10 THEN AA=AA+1
640 IF A=11 THEN AB=AB+1
650 IF A=12 THEN AC=AC+1
660 IF A=13 THEN AD=AD+1
670 PRINT"[CLEAR]PUNTI:";A:PRIN
    T"SCHEDA N.";SC:PRINT"COLON
    NA N.";CO
680 PRINT"[DOWN][RVS]PREMI UN T
    ASTO PER CONTINUARE"
690 IF FI THEN 710
700 GOTO 780
710 PRINT"[CLEAR]RIASSUNTO:":PR
    INT"[3 DOWN]PUNTI 10:"AA:PR
    INT"PUNTI 11:"AB
720 PRINT"PUNTI 12:"AC:PRINT"PU
    NTI 13:"AD
730 FOR J=1 TO 6:PRINTCHR$(17):
    NEXT
740 PRINT"[RVS]R=[RVOFF]RESTART
              [RVS]E=[RVOFF]USCI
    TA DAL PROGRAMMA"
750 GET A$:IF A$="R" THEN 820:R
    EM RIPRENDE LAVORO
760 IF A$="E" THEN PRINT"[CLEAR
    ]";:END :REM FINE L
770 GOTO 750
780 GET A$:IF A$="" THEN 780
790 PRINT"[CLEAR]":SYS8735
800 GOTO 580
810 FOR J=1 TO 126:READ A:POKE
    8723+J,A:NEXT:RETURN
820 ZX=1:AA=0:AB=0:AC=0:AD=0:FI
    =0:GOTO 200
830 REM  DATI L/M
840 HK=0:FOR J=1 TO 100:READ A:
    HK=A+HK:POKE 8723+J,A:NEXT
850 IF HK<>9714 THEN PRINT"ERRO
    RE NEI DATA":END
860 RETURN
870 DATA  169,120,133,38,169,34
    ,133,39,76,41,34,173,18,34,
    133,38
880 DATA  173,19,34,133,39,160,
    0,140,15,34,162,1,177,38,22
    1,255
890 DATA  33,240,59,230,38,208,
    2,230,39,232,224,14,208,238
    ,238,16
900 DATA  34,208,3,238,17,34,17
    3,16,34,205,13,34,208,9,173
    ,17
910 DATA  34,205,14,34,208,1,96
    ,173,15,34,201,9,208,3,76,4
    1
920 DATA  34,144,194,165,38,141
    ,18,34,165,39,141,19,34,96,
    238,15
930 DATA  34,76,55,34
```

C 64 - C 128

Riduzione di sistemi condizionati

Un sistema per ridurre al minimo il numero di colonne da giocare al Totocalcio

di Antonio Pastorelli

Ed eccoci ad un altro appuntamento con i giochi d'azzardo, tra i quali includiamo il totocalcio dal momento che decine di migliaia di persone, ricorrendo all'uso intensivo dei sistemi, sperperano settimanalmente centinaia di migliaia di lire.

In questo articolo daremo un ulteriore strumento per fare giocate più efficienti, cioè per spendere poco ma bene.

Vi proponiamo infatti un programma per ridurre i sistemi condizionati (vedi programma pubblicato su Commodore Computer Club N.35).

• Innanzitutto si prepara il sistema-base integrale.
• Si carica e si "lancia", quindi, il programma TOT 13 pubblicato su CCC di ottobre.
• Si pongono i condizionamenti in modo da eliminare le colonne che, secondo voi, hanno una bassissima probabilità di vincita.
• Finita la selezione delle colonne (con TOT 13), alla domanda "Registri" si risponde affermativamente, realizzando in tal modo un archivio su nastro o disco.
• Si carica il programma di queste pagine e si impartisce il RUN.

Questo programma caricherà, una alla volta, le colonne precedentemente registrate, e selezionerà quelle da giocare.

Vediamo ora in base a quale criterio avviene la selezione.

La prima colonna caricata in memoria è anche la prima colonna valida ai fini della riduzione.

Per ognuna delle successive, invece, il computer esegue un controllo tra queste e tutte quelle precedente-

mente selezionate come valide.

Se dal confronto risulta che tra le colonne valide ne esiste almeno una che differisce da quella caricata dall'archivio anche per un solo pronostico, allora non viene giocata; se fosse quella vincente, infatti, si totalizzerebbe comunque almeno un 12.

Negli altri casi la colonna viene considerata valida ed inserita nel gruppo delle colonne da giocare.

Il programma TOT 13, per come è stato pubblicato sul N.35, genera un file chiamato "TOT". Poichè su un disco non possono esservi due file con lo stesso nome potrebbe darsi il caso che si abbia un dischetto con più file generati da TOT 13 (con nomi diversi grazie a modifiche del programma da parte del lettore o a successive operazioni di Rename). Chi possiede il drive, pertanto, si assicuri che il file dal quale vuole ricavare la riduzione sia registrato effettivamente con il nome TOT; in caso contrario bisognerà cambiarlo con il comando Rename.

Terminata la selezione il computer chiede se si desidera registrarla oppure visualizzarla subito.

Nel primo caso viene creato un archivio, su nastro o disco, chiamato "R.TOT". Nell'altro, invece, le colonne vengono stampate su video o stampante (rispondendo all'apposita richiesta con "V" oppure con "S").

Se si sceglie di registrare, l'archivio potrà essere successivamente letto con il programma DISPLAY, pubblicato insieme con TOT 13, avendo l'accortezza di cambiare il nome del file da TOT a "R.TOT", nelle linee 160 e 170.

Per verificare che il programma funzioni correttamente, ricorriamo ad un esempio banale ma utilissimo.

Dopo aver caricato TOT 13, impostate un sistema formato da quattro triple (primi quattro pronostici) e nove fisse (i rimanenti).

Alla domanda sui segni, impostate la seguente direttiva:

1, da 10 a 13
X, da 1 a 3
2, da 1 a 2

Alle domande sulla consecutività dei segni rispondete nel modo seguente:

1, massimo=10
X, massimo=2
2, massimo=2

Se avete digitato correttamente il listato di CCC N.35, il computer dovrebbe elaborare 34 colonne.

Nel digitare righe di programma basic che contengono istruzioni DATA, è piuttosto facile incorrere in errori di digitazione. Supponiamo che un'ipotetica linea basic numerata con 1200 debba contenere i tre valori: 123, 456, 789. Ecco alcuni esempi di errori più frequentemente commessi:

```
1200 DATA,123,456,789
```

C'è una virgola dopo la parola "DATA". I dati letti dal computer sono, in questo caso, quattro: 0, 123, 456, 789. Se, infatti, non figura alcun carattere dopo l'istruzione DATA, automaticamente viene assunto il valore nullo (0).

```
1200 DATA 123,456,789,
```

In questo caso, dopo il numero 789, il computer, grazie alla presenza della virgola erroneamente inserita, "crede" che ci sia un altro valore e, non trovandolo, lo assume come nullo (0).

```
1200 DATA 1234,56,789
```

La virgola è posizionata male, vale a dire dopo il carattere "4" e non dopo il carattere "3". Il computer non può sapere se il valore esatto è 123 oppure 1234 e individuare un errore, in questo caso, risulta piuttosto laborioso.

Queste colonne, elaborate con il programma di queste pagine, dovrebbero ridursi alle seguenti dieci (in cui, per semplicità, sono stati omessi i pronostici "fissi"):

```
1   11X 211 111 1111
2   112 X11 111 1111
3   1X1 211 111 1111
4   1X2 111 111 1111
5   121 X11 111 1111
6   12X 111 111 1111
7   X11 211 111 1111
8   X12 111 111 1111
9   211 X11 111 1111
10  21X 111 111 1111
```

La Posta del Totocalcio

Alcuni lettori hanno fatto notare che il programma TOT 13 non gira sul C/128. In effetti ciò è vero e a nostra discolpa possiamo solo dire che una precedente versione, che girava perfettamente sul C/128, ci ha indotto ad affermare che il programma era idoneo anche per tale computer.

In seguito, però, volendo rendere valido il listato anche per il Plus/4, è stato modificato rendendolo, di fatto, incompatibile con il C/128. Nell'articolo, purtroppo, abbiamo dimenticato di accennare a tale limitazione. Non dimentichiamo, però, che chi possiede il C/128 possiede "anche" il C/64 ed il programma gira perfettamente sul C/128 attivato in modo 64.

Altri lettori, che intendono utilizzare il registratore a cassette per registrare i dati di TOT 13, segnalano un "Illegal Device Number Error" alla linea 730.

Ciò capita se si utilizzano particolari TOOL, tra cui lo Speed Dos e altre cartucce che, in genere, aumentano la velocità del drive.

Molto spesso, infatti, alcuni Tool inibiscono l'uso dei file su cassetta, facendo comparire il messaggio di errore citato nel caso in cui si aprano file verso il registratore.

In questi casi, quindi, o si utilizza il drive oppure si elimina la cartuccia prima di caricare ed utilizzare il programma TOT 13.

```
100 REM DISPLAY PER TOT 13
120 DIM A$(700):C$=CHR$(147):D$
    =CHR$(17):E$=CHR$(157):F$=C
    HR$(29)
130 FOR I=1 TO 17:G$=G$+CHR$(17
    ):NEXT:H$=CHR$(19)
140 PRINTC$"D I S P L A Y"
145 PRINT:INPUT "NOME DEL FILE"
    ;W0$
150 PRINT:PRINT"I DATI SONO SU
    NASTRO O DISCO? (N/D)"
160 GET A$:IF A$="N" THEN OPEN
    1,1,0,W0$:GOTO 190
170 IF A$="D" THEN OPEN 1,8,12,
    W0$+",S,R":GOTO 190
180 GOTO 160
190 INPUT#1,NR
200 PRINTC$"LETTURA DATI"
210 FOR J=1 TO NR:FOR H=1 TO 13
    :INPUT#1,A$:A$(J)=A$(J)+A$:
    NEXTH,J:CLOSE 1
220 PRINTC$"OK"
230 PRINT"PREMI UN TASTO PER IN
    IZIARE"
240 GET A$:IF A$="" THEN 240
250 H=1
260 PRINTC$"DA";H;"A";H+19:P=0
270 PRINT:FOR J=H TO H+19
280 FOR K=1 TO 25 STEP 2:A$=MID
    $(A$(J),K,1)
290 IF A$="1" THEN 320
300 IF A$="2" THEN A$="X"
310 IF A$="3" THEN A$="2"
320 PRINTA$;D$;E$;:NEXTK
330 PRINTH$:PRINT:PRINT TAB(J);
340 NEXTJ:PRINTH$;G$"*:PROSSIMO
     GRUPPO":PRINT"[3 UP]↑";
350 GET Z$:IF Z$<>"*" THEN 410
360 IF Z$="" THEN 350
370 H=H+20
380 IF H<=NR THEN 260
390 PRINT:PRINT:PRINT:PRINT"FIN
    E"
400 END
410 IF Z$=CHR$(29) THEN P=P+1:G
    OTO 440
420 IF Z$=CHR$(157) THEN P=P-1:
    GOTO 460
430 GOTO 350
440 IF P<=19 THEN PRINT"[LEFT]
    ↑";:GOTO 350
450 P=19:GOTO 350
460 IF P>=0 THEN PRINT"[LEFT] [
    2 LEFT]↑";:GOTO 350
470 P=0:GOTO 350
```

```
110 REM RIDUZIONE DI SISTEMI
120 REM CONDIZIONATI
140 REM PER C/64
160 REM BY ANTONIO PASTORELLI
180 :
190 FOR J=1 TO 40:Z$=Z$+CHR$(32
    ):NEXT
200 POKE 55,50:POKE 56,33:POKE
    687,1:POKE 688,0:POKE 691,2
    07:POKE 692,33
210 POKE 685,220:POKE 686,33
220 PRINT"[CLEAR][RVS]   RIDUZI
    ONE DI SISTEMI CONDIZIONATI
       "Z$
230 GOSUB 720:REM PREPARA ROUTI
    NE L/M
240 PRINTCHR$(17)"L'ARCHIVIO SI
     TROVA SU NASTRO O SU DISCO
    ?"
250 GET A$:IF A$="N" THEN SA=1:
    GOTO 275
260 IF A$="D" THEN SA=8:GOTO 27
    5
270 GOTO 250
275 INPUT "NOME FILE:";W0$:IF W
    0$="" THEN PRINT"[UP]";:GOT
    O 275
280 REM APRE FILE IN LETTURA
285 IF SA=8 THEN W0$=W0$+",S,R"
    :GOTO 300
290 OPEN 1,1,0,W0$:GOTO 310
300 OPEN 1,8,3,W0$
310 INPUT#1,NR:PRINT"[CLEAR]
       ATTENDI:"NR
320 FOR J=0 TO 12:INPUT#1,A:POK
    E 8655+J,A:NEXT:REM LEGGE P
    RIMA COLONNA
330 FOR J=2 TO NR:PRINT"[HOME]"
    J
340 FOR X=0 TO 12:INPUT#1,A:POK
    E 8500+X,A:NEXTX:REM LEGGE
    COLONNE DAL FILE 'TOT'
350 SYS8533:REM CONTROLLO
360 NEXTJ:CLOSE 1
370 CL=PEEK(687)+256*PEEK(688):
```

```
    REM NUMERO COLONNE VALIDE
380 PRINT"[CLEAR]HO FINITO LA R
    IDUZIONE."
390 PRINT"LE COLONNE SELEZIONAT
    E SONO:"CL
400 REM SCELTE
410 PRINTCHR$(17):PRINT"1)REGIS
    TRA":PRINT"2)DISPLAY":PRINT
    "FINE LAVORO"
420 PRINTCHR$(17)"CHE COSA SCEG
    LI?"
430 GET A$:IF A$="1" THEN 600
440 IF A$="2" THEN 470
450 IF A$="3" THEN END
460 GOTO 430
470 PRINT"[CLEAR]SU [RVS]V[RVOF
    F]IDEO O [RVS]S[RVOFF]TAMPA
    NTE?"
472 GET A$:IF A$="" THEN 472
474 DV=3:IF A$="S" THEN DV=4
475 OPEN 1,DV
478 PRINT"[CLEAR][RVS]'S'[RVOFF
    ] PER FERMARE LO SCROLLING"
    :PRINT
480 REM STAMPA RISULTATI SU VID
    EO
490 FOR J=1 TO CL:PRINT#1, TAB(
    3-LEN(STR$(J)))CHR$(18)JCHR
    $(146)" ";
500 FOR X=1 TO 13:A=PEEK(8641+1
    3*J+X)
510 REM CONVERSIONE SEGNI
520 IF A=1 THEN A$="1"
530 IF A=2 THEN A$="X"
540 IF A=3 THEN A$="2"
550 PRINT#1,A$;:IF X=3 OR X=6 O
    R X=9 THEN PRINT#1," ";
560 NEXTX
570 GET B$:IF B$="S" THEN 590
580 PRINT#1:NEXTJ:PRINT#1:CLOSE
     1
585 PRINT"[DOWN][RVS]PREMI UN T
    ASTO":GOSUB 690:GOTO 380
590 GOSUB 690:GOTO 580
600 IF SA=8 THEN 635
610 PRINT"[CLEAR]POSIZIONA IL N
    ASTRO SU UNO SPAZIO LIBERO"
    :GOSUB 690
620 REM APERTURA FILE 'R.TOT' I
    N SCRITTURA
630 OPEN 1,1,1,"R.TOT":GOTO 650
635 W0$="R."+W0$:W0$=LEFT$(W0$,
    LEN(W0$)-1)+"W"
640 PRINT"INSERISCI IL DISCO NE
    L DRIVE":GOSUB 690:OPEN 1,8
    ,12,W0$
650 PRINT#1,CL
660 FOR J=1 TO CL
670 FOR X=1 TO 13:PRINT#1,PEEK(
    8641+X+J*13):NEXTX
680 NEXTJ:CLOSE 1:END
690 GET B$:IF B$="" THEN 690
700 RETURN
710 REM LETTURA DATI ROUTINE L/
    M
720 FOR J=0 TO 137:READ A:KK=KK
    +A:POKE 8513+J,A:NEXT:RETUR
    N
722 IF KK<>14619 THEN PRINT"ERR
    ORE NEI DATA.":END
725 RETURN
730 REM DATI ROUTINE L/M
740 :
750 DATA  160,0,185,52,33,209,2
    ,208,6,200,192,13,208,244,9
    6,230
760 DATA  4,76,74,33,173,175,2,
    174,176,2,141,177,2,142,178
    ,2
770 DATA  173,179,2,174,180,2,1
    33,2,134,3,169,0,133,4,32,6
    5
780 DATA  33,165,4,201,1,208,1,
    96,206,177,2,173,177,2,201,
    255
790 DATA  208,3,206,178,2,173,1
    77,2,208,50,173,178,2,208,4
    5,238
800 DATA  175,2,208,3,238,176,2
    ,173,173,2,174,174,2,133,2,
    134
810 DATA  3,160,0,185,52,33,145
    ,2,200,192,13,208,246,173,1
    73,2
820 DATA  24,105,13,141,173,2,1
    44,3,238,174,2,96,165,2,24,
    105
830 DATA  13,133,2,144,2,230,3,
    76,107,33
840 END
```

Quale fascicolo manca alla tua enciclopedia Commodore?

Per ordinare i fascicoli mancanti alla tua collezione di Commodore Computer Club utilizza l'apposita scheda in fondo alla rivista.

ALTA RISOLUZIONE A BASSO COSTO
LA MIGLIORE PERIFERICA PER GRAFICA

GRAFPAD II

Software compreso su cassetta e disco

- DIMENSIONE DISEGNO: FORMATO A4
- ALTA RISOLUZIONE A COLORI
- PER CASA E UFFICIO
- DIVERSI PROGRAMMI OPTIONAL
- DISEGNO A MANO LIBERA
- DISEGNO CIRCUITI ELETTRICI
- CREAZIONE DI BIBLIOTECA SIMBOLI GRAFICI

LA PRIMA TAVOLETTA GRAFICA A BASSO COSTO CHE OFFRE
LE PRESTAZIONI E DURABILITA' RICHIESTE DALLE APPLICAZIONI
INDUSTRIALI, AZIENDALI, SCOLASTICHE ECC.
E' PICCOLA, PRECISA E AFFIDABILE.

PER AMSTRAD 464-664-6128
PER COMMODORE 64-128-128D

NON HA BISOGNO DI MANUTENZIONE

NOVITA' ASSOLUTA PER COMMODORE 64
RICONOSCITORE VOCALE: comanda a voce il tuo Commodore 64 tramite microfono

NOVITA' ASSOLUTA IN ITALIA
Televisore TASCABILE: seguite le trasmissioni televisive in qualsiasi luogo. Dimensioni: 13 cm x 7 cm x 3 cm.

AMSTRAD 464-6654-6128

HARD

Penna ottica
Espansione di memoria 64K - 256K
Sintetizzatore vocale
Disc Drive con controller
Stampante DMP2000

TASWORD: WP potente per creazione di testi e documenti
TASPRINT: Programma supplementare al precedente per la stampa
TASCOPY: Hardcopy-stampa immagini anche in formato poster di tutto ciò che compare su video
MASTERFILE: Sistema di archiviazione e ricerca selettiva - potente DATABASE
MUSIC-SYSTEM: Per comporre musica

AMSTRAD PCW 8256 -8512

HARD

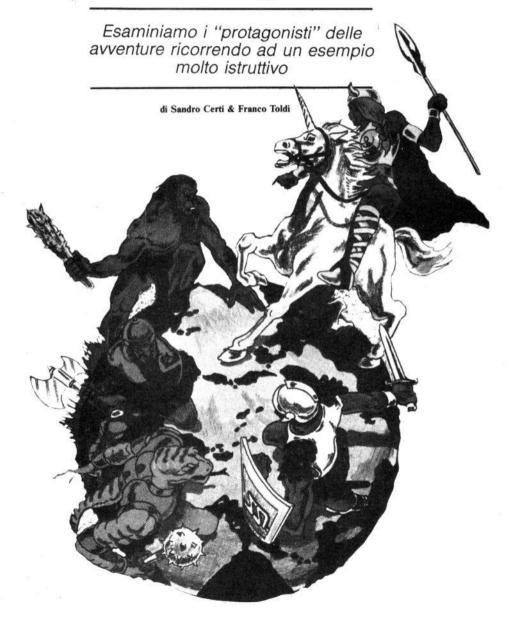
Espansione di memoria 256KB + secondo disco da 1 Megabyte in kit di montaggio
GRAFPAD III: tavoletta grafica ad alta risoluzione per CAD professionale completa di software e manuali in italiano

TASWORD 8000: elaborazione testi con abbinamento testi a indirizzi, stampa etichette, stampa in protocollo
TASPRINT 8000:Programma complementare al precedente per stampa professionale con 8 stili diversi
CYRUS II: scacchi tridimensionali professionali

CONSEGNA IN TUTTA ITALIA: TELEFONARE PER INFORMAZIONI
S.T. Syscom - Via B. Palazzo, 13/B - 24100 Bergamo - Tel. 035/239751

Laser, pistole, spade e gemme

Esaminiamo i "protagonisti" delle avventure ricorrendo ad un esempio molto istruttivo

di Sandro Certi & Franco Toldi

Il cammino dell'avventuriero è letteralmente cosparso di oggetti: alcuni hanno solo una funzione decorativa, limitandosi ad arredare il mondo del gioco, altri sono utili, spesso indispensabili, per superare gli ostacoli e le trappole che ci attendono sul cammino.

Nel precedente articolo abbiamo visto come creare i luoghi in cui si svolge l'avventura. Abbiamo poi allestito le routine necessarie per permettere al giocatore di muoversi all'interno del mondo così creato.

In questo articolo esamineremo le routine che consentiranno di collocare i mezzi ritenuti necessari nei luoghi scelti. Vedremo poi le classiche routine "prendere", "lasciare" e "inventario", che consentiranno al nostro giocatore di avvalersi dei mezzi a sua disposizione prelevandoli da un certo luogo, portandoli con sè, verificandone durante la via il possesso e depositandoli nel luogo opportuno.

Per prima cosa vedremo come dotare alcuni luoghi dei mezzi necessari per affrontare gli ostacoli che lo attendono. Parleremo dei soli "ostacoli", cioè di quei luoghi dove -per il superamento di una difficoltà- si rendono necessari mezzi appropriati. Si tratta di oggetti che l'avventuriero può tuttavia reperire percorrendo a ritroso il cammino fatto. Non parleremo invece delle "trappole", quei luoghi cioè ove il possesso di un oggetto adeguato è talmente essenziale per l'avventuriero, bloccato nella trappola, che la sua mancanza ne determina spesso la morte. Parleremo delle trappole prossimamente, e ne approfitteremo, magari, per introdurre altri elementi dell'organizzazione del gioco.

Le routine di dispiegamento dei mezzi

Nell'articolo precedente abbiamo già fatto uso sia di variabili numeriche che di variabili stringa. Prendevamo tuttavia in considerazione un solo valore numerico o una sola stringa di caratteri per volta da assegnare alla variabile interessata.

Con il termine "mezzi" intendiamo far riferimento ad un insieme di valori numerici, o di strighe di caratteri, organizzati in base ad un principio ordinatore.

Nel programma dimostrativo abbiamo pensato a due soli oggetti collocati in altrettanti luoghi precisi. Vogliamo cioè che si stabilisca un legame tra il luogo dove si trova un oggetto, il nome dello stesso e la sua descrizione.
Esempio:

Nel luogo N.4 c'è una lunga spada artisticamente decorata.

Nel luogo N.8 c'è un piccolo sacchetto di riso.

Raccolte di dati, ordinati come descrito, sono comunemente chiamate "array".

Come per le altre variabili, a ciascun array va assegnato un nome seguito dal simbolo di percentuale (%) se contenente valori numerici, "$" se contenente stringhe.

Nel nostro caso abbiamo i tre arrays:

OG%() OG$() Do$()

Ci si riferisce in genere ai singoli elementi dell'array con il nome dell'array seguito da un valore numerico, tra parentesi, indicante il posto che l'elemento occupa nella tabella. Seguendo l'esempio appena visto, OG%(2) indicherà il luogo n.8, OG$(1) l'oggetto spada e DO$(2) il piccolo sacchetto di riso.

Pensando che a partire dal nostro programma ne verranno costruiti di più estesi, abbiamo provveduto a dimensionare gli array del dimostrativo con l'apposita istruzione DIM.

I commenti alle routine.

Ricordiamo, anzitutto, che solo un paio di Poke (relative ai colori dello schermo) impediscono la compatibilità totale con computer diversi dal C/64: sarà sufficiente non trascriverle per veder girare il programma su qualsiasi calcolatore.

Riteniamo utile segnalare che, pur mantenendo la denominazione completa del mezzo nella variabile OL$ (I), abbiamo messo in opera la variabile OG$(I) che di tale denominazione contiene solo le prime tre lettere. Ciò al fine -come vedremo più avanti- di velocizzare il lavoro del parser.

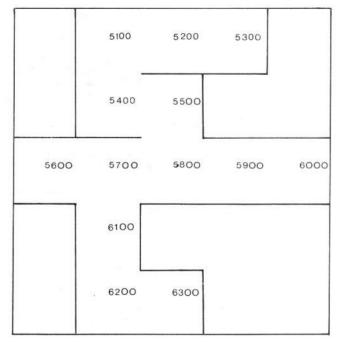
Le routine di movimentazione dei mezzi: un primo parser.

Nel programma dimostrativo del precedente articolo era già all'opera un minuscolo parser o analizzatore di linguaggio. Nelle righe 800-900 (vedi CCC N.37) esaminava gli input del giocatore, provvedendo in caso di istruzione corretta ad operare l'opportuno cambiamento di luogo e, altrimenti, ad informare il giocatore che la direzione scelta non era praticabile.

Per "prendere" un oggetto da un luogo e "lasciarlo" in un altro abbiamo bisogno di istruzioni costituite da almeno un verbo ed un nome. Siamo cosìdi fronte al classico parser a due parole dei primi Adventure Game. A parser più complessi in grado di analizzare due righe o più di imput, sul modello di quello impiegato dalla Infocom con Zork, provvederemo con altri articoli.

Per una più facile comprensione delle routine che seguono, ricordiamo qui il lavoro che svolgono le funzioni Len(stringa) e Mid$(stringa) che in esse utilizzeremo.

La funzione Len(stringa) fornisce in termini numerici la lunghezza di una stringa alfanumerica, cioè di quanti caratteri è composta. La funzione Mid$(stringa (prima epressione numerica), (seconda espressione numerica)) preleva dalla stringa di caratteri una sottostringa. Prende come carattere di inizio quello indicato dalla prima espressione numerica e ne aggiunge tanti quanti sono stati indicati nella seconda.

Così, ad esempio, nel caso della stringa "ALABARDA" composta da otto caratteri, il risultato di MID$ ("ALABARDA",1,3) sarà "ALA", di MID$("ALABARDA,4,5) "BARDA".

Routine A: righe 100/140
La routine si limita a smistare i comandi di una lettera -relativi ad inventario, aiuto e movimentazione- alle righe di programma dove essi vengono interpretati fornendo, se il caso, le risposte opportune. Si segnala qui il rimando alla routine E, inventario, direttamente connessa con le routine di movimentazione.

Routine B: righe 390/410
Questa routine, non prevedendo il nostro progetto comandi di una parola, provvede ad inibirli, informando il giocatore della cosa.

Routine C: righe 1090/1140
Questa routine è il centro del parser impiegato in questo articolo. Avvalendosi delle funzioni LEN e MID$, separa l'imput del giocatore in due parti. Denomina "verbo" la prima e "nome" la seconda, rendendole disponibili per ulteriori confronti.

Routine D: righe 1190/1220
Prosegue il lavoro iniziato dalla precedente, confrontando il verbo dell'imput con quelli previsti dal gioco. In caso positivo l'analisi viene fatta proseguire smistando alle specifiche routine dei verbi. In caso negativo il giocatore viene informato della cosa.

Routine E: righe 1290/1350
Fornisce, dietro richiesta, l'inventario degli oggetti che il giocatore reca con sè. Questa routine è strettamente legata alle seguenti "prendi" e "lascia".

Routine F: righe 1390/1430
Fornisce, nei luoghi a ciò designati, elementi che possono aiutare il giocatore o metterlo fuori strada.

Routine G: righe 1990/2070
In seguito all'istruzione del giocatore di prendere un determinato oggetto, lo informa sulla "situazione". L'oggetto richiesto può, o meno, essere preso (facendo ad esempio parte dell'arredamento del luogo) o essere già in possesso del giocacatore o non essere nel luogo in questione.

Routine H: righe 2090/2170
La routine H, a fronte di un ordine del giocatore di lasciare un determinato oggetto, lo informa sulla "situazione". L'oggetto può, o meno, essere lasciato o non essere in possesso del giocatore o essere lasciato nel luogo scelto.

Evidenziamo che nel caso specifico della routine del programma dimostrativo, le linee 2150 e 2160 sono usate per simulare il superamento dei due ostacoli.

Programma dimostrativo

Il programma pubblicato ha lo scopo di permettervi di utilizzare tutte le routine fin qui descritte.

E' stato concepito sotto forma di gioco per evidenziare l'avvicinamento che in esso si compie ad una avventura vera e propria.

Siamo partiti dall'idea di un tempio in cui l'avventuriero si trovasse rinchiuso per ignoti motivi. Abbiamo quindi pensato che, nel suo percorso verso la libertà, incontrasse due ostacoli superabili mediante il possesso e l'uso di due oggetti.

In un primo approccio al problema della sceneggiatura dei giochi di avventura facciamo rilevare che i due ostacoli sono tra loro strettamente legati. Il superamento dell'uno è necessaria premessa a quello dell'altro.

```
1 REM  DIMOSTRATIVO CCCADVPRG
   2
2 REM  MEZZI:MOVIMENTAZIONE
20 REM  COLORE SCHERMO E CORNI
   CE
21 POKE  53280,9:POKE 53281,9
30 REM  SCRIVE MINUSCOLO
31 POKE 53272,23
100 REM  INIZIALIZZ. VAR. MEZZI
110 DIM  OG%(2),OG$(2),DO$(2),O
   L$(2)
120 FOR I=1 TO 2:READ OG%(I),OG
```

Hard & soft

LA NIWA PUÒ ESSERE LA TUA MIGLIORE AMIGA®

Distributore autorizzato COMMODORE

**In regalo a tutti gli acquirenti di un PC AMIGA
la tessera del NIWA AMIGA CLUB.
AMIGA costa £ 2.200.000 + IVA
consegna GRATIS IN TUTTA ITALIA.
Tutto il software disponibile
e l'hardware novità.**

Inoltre la NIWA vi propone per il vostro C/64-C/128:

Floppy disk "Memorette" 5¹/⁴ ssdd 100% error free .... cd.	L. 1.300
Floppy disk bulk 3¹/² dsdd 100% error free .... da	L. 3.800
O.M.A. Non permettere che i tuoi programmi originali si ROVININO. Con O.M.A., puoi fare una copia di sicurezza in un unico file (!) ricassettabile del tuo software su disco o su nastro	L. 99.000
HACKER Cartridge: trasferisce il 99% del tuo software protetto da nastro e da disco a disco in soli 4 minuti senza bisogno di conoscenza Linguaggio. Macchina e di compattamento; lavora inoltre senza disco di lavoro e può essere attivata a programma già caricato. 128 compatibile (in modo 64)	L. 80.000
HACKER-TAPE: permette di ricassettare qualsiasi tipo di programma precedentemente trattato con HACKER, senza nessun problema di blocchi, leggendo in turbo da disco e scrivendo in turbo su nastro	L. 45.000
OFFERTA: HACKER + HACHER TAPE	L. 99.000
MPS 802 New Graphic CON MONTAGGIO GRATUITO rende 100% compatibile la tua MPS 802 con i programmi di Grafica	L. 80.000

Speeddos per C64 **L. 65.000** per C128 **L. 85.000,** Fast load reset **L. 35.000**, Isepic **L. 50.000**, Capture **L. 99.000.**

```
    $(I),DO$(I),OL$(I):NEXT
130 DATA 4,SAC,UN PICCOLO SACCH
    ETTO DI RISO,SACCHETTO
140 DATA 8,SPA,UNA LUNGA SPADA
    ARTISTICAMENTE DECORATA,SPA
    DA
190 REM  FISSA IL LUOGO INIZIAL
    E
200 P=11
290 REM  SMISTA SULLE LINEE DES
    C.LUOGHI
300 IF P<6 THEN ON P GOTO 0,510
    0,5200,5300,0
310 IF P<11 THEN ON P-5GOTO 0,5
    400,5500,0,0
320 IF P<16 THEN ON P-10GOTO 56
    00,5700,5800,5900,6000
330 IF P<21 THEN ON P-15GOTO 0,
    6100,0,0,0,
340 IF P<26 THEN ON P-20GOTO 0,
    6200,6300,0,0
390 REM  SCRIVE GLI OGGETTI NEI
     LUOGHI
400 FOR I=1 TO 2:IF OG%(I)=P  T
    HEN  PRINTDO$(I)
410 NEXT
590 REM  SCRIVE LE DIREZIONI PE
    RCORRIBILI
600 PRINT"PUOI ANDARE A:"
610 IF N>0  THEN  PRINT "NORD
    ";
620 IF S>0  THEN  PRINT "SUD
    ";
630 IF E>0  THEN  PRINT "EST
    ";
640 IF O>0  THEN  PRINT "OVEST
    "
680 REM  RICHIEDE ISTRUZIONI AL
     GIOC.
690 PRINTCHR$(13)
700 INPUT "COSA FAI";A$
710 REM  COMANDI DI UNA LETTERA
720 IF  A$="I" OR A$="A" THEN 8
    00
730 IF  A$="N" OR A$="S" OR A$=
    "O" OR A$="E" THEN  800
740 GOTO 1000
790 REM  PARSER INVENTARIO,AIUT
    O,DIREZIONI
800 IF  A$="I" THEN  1290
810 IF  A$="A" THEN  1390
820 IF  A$="N"  AND  N>0  THEN
     P=N:GOTO 300
830 IF  A$="S"  AND  S>0  THEN
     P=S:GOTO 300
840 IF  A$="E"  AND  E>0  THEN
     P=E:GOTO 300
850 IF  A$="O"  AND  O>0  THEN
     P=O:GOTO 300
890 REM  DIREZIONI NON PRATICAB
    ILI
900 PRINT"NON PUOI ANDARE IN QU
    ESTA DIREZIONE!":GOTO 600
1000 REM  INIBISCE I COMANDI DI
     UNA PAROLA
1010 FOR I=1 TO LEN(A$)
1020 IF  MID$(A$,I,1)=" " THEN 1
     100
1030 NEXT
1040 PRINT"USA ISTRUZIONI DI DUE
      PAROLE":GOTO 690
1090 REM  PARSER: ISTRUZIONI DI
     DUE PAROLE - SEPARA VERBI E
      NOMI-
1100 FOR  SP=1 TO  LEN(A$)
1110 IF MID$(A$,SP,1)=" " THEN
     1130
1120 NEXT
1130 VE$=MID$(A$,1,3)
1140 NO$=MID$(A$,(SP+1),3)
1190 REM  PARSER: ESAMINA I VERB
     I
1200 IF VE$="PRE" OR VE$="ALZ" O
     R VE$="SOL"  THEN  2000
1210 IF VE$="LAS" OR VE$="POS" O
     R VE$="DEP"  THEN  2100
1220 PRINT"NON CAPISCO COSA VUOI
      FARE!":GOTO 690
1290 REM    ROUTINE INVENTARIO
1300 PRINT"L'INVENTARIO DEGLI OG
     GETTI CHE HAI CON TE E':":I
     N=0
1310 FOR I=1 TO 2
1320 IF OG%(I)=-1  THEN  PRINTOL
     $(I):IN=IN+1
1330 NEXT
1340 IF IN=0 THEN  PRINT"NULLA"
1350 GOTO  690
1390 REM    ROUTINE AIUTO
1400 IF P=2 THEN  PRINT"NESSUN A
     IUTO!"
```

NOVITA'

Hard & soft
NIWA

KIT 802
NEW GRAPHIC

Finalmente la tua stampante
è 100% compatibile con tutti
i programmi di grafica.
(Geos 12 - Newsroom - Print Shop - Print Master)

Kit di montaggio completo di istruzioni
in italiano montaggio gratuito immediato
agli acquirenti che porteranno la stampante
presso il NS negozio

L. 80.000

O.M.A. Cartridge

Non permettere che i tuoi programmi
originali si rovinino.
Con O.M.A., puoi fare una copia di sicurezza
in un unico file (!!) (ricassettabile)
del tuo software su disco o nastro.

solo **L. 99.000**
(Manuale in italiano)

Spedizioni contrassegno in tutta Italia - Si accettano ordini telefonici
Niwa Via Buozzi 94 - P.O. BOX 83 20099 Sesto S. Giovanni (MI) MM. Marelli - Tel. 02/2440776 - 2476523

128 KBYTES

SINCLAIR COM

COMMODORE

+

MSX

=

```
1410 GOTO 690
1420 IF P=14 THEN  PRINT"MA VA!!
     "
1430 GOTO 690
1990 REM    ROUTINE "PRENDI"
2000 FOR I=1 TO 2
2010 IF OG$(I)=NO$ THEN 2040
2020 NEXT
2030 PRINT"NON PUOI!":GOTO 690
2040 IF OG%(I)=-1 THEN  PRINT"L'
     HAI GIA' PRESO!":GOTO 600
2050 IF OG%(I)<>P THEN  PRINT"/O
     N E' QUI!":GOTO  600
2060 PRINT"VA BENE!":OG%(I)=-1
2070 GOTO 600
2090 REM    ROUTINE "LASCIA"
2100 FOR I=1 TO 2
2110 IF OG$(I)=NO$ THEN 2130
2120 NEXT
2125 PRINT"NON PUOI!":GOTO 600
2130 IF OG%(I)<>-1 THEN  PRINT"N
     ON L'HAI CON TE!":GOTO 600
2140 PRINT"VA BENE!":OG%(I)=P
2150 IF  NO$="SPA" AND P=2 THEN
     P=3:GOTO 300
2160 IF  NO$="SAC" AND P=14 THEN
      P=15:GOTO 300
2170 GOTO 690
5099 REM  LUOGO 2
5100 PRINT"[CLEAR]AD EST SULLA P
     ORTA UNA GRANDE STATUA      "
     ;
5110 PRINT"DEL DIO. UNA MANO E'
     PROTESA. SEMBRA AT-";
5120 PRINT"TENDERE QUALCOSA."
5130 N=0:S=7:O=0:E=0:GOTO 390
5199 REM  LUOGO 3
5200 PRINT"[CLEAR]UN BASSO CORRI
     DOIO RICOPERTO DI LEGNO.    "
5210 N=0:S=0:O=2:E=4:GOTO 390
5299 REM  LUOGO 4
5300 PRINT"[CLEAR]UNA GRANDE STA
     TUA DEL DIO. SULLA MANO     "
     ;
5310 PRINT"DESTRA VARI OGGETTI.
     UN VASO DI PERLE.   "
5320 N=0:S=0:O=3:E=0:GOTO 390
5399 REM  LUOGO 7
5400 PRINT"[CLEAR]LA PICCOLA SAL
     A E' VUOTA."
5410 N=2:S=0:O=0:E=8:GOTO 390
5499 REM  LUOGO 8
5500 PRINT"[CLEAR]UNA PICCOLA ST
     ANZA. LUNGHI MANTELLI. AR-"
     ;
5510 PRINT"MATURE."
5520 N=0:S=13:O=7:E=0:GOTO 390
5599 REM  LUOGO  11
5600 PRINT"[CLEAR]UNO STRETTO CO
     RRIDOIO. LA GRANDE PORTA   "
     ;
5610 PRINT"AD OVEST E' CHIUSA PE
     R SEMPRE."
5620 N=0:S=0:O=0:E=12:GOTO 390
5699 REM  LUOGO 12
5700 PRINT"[CLEAR]UNA PICCOLA ST
     ANZA RICAVATA NELLA ROCCIA"
5710 PRINT". TRE PASSAGGI."
5720 N=0:S=17:O=11:E=13:GOTO 390
5799 REM  LUOGO 13
5800 PRINT"[CLEAR]UNA GRANDE SAL
     A DOMINATA DAL DIO SHIVA. "
5810 N=8:S=0:O=12:E=14:GOTO 390
5899 REM  LUOGO 14
5900 PRINT"[CLEAR]CONTRO LA PARE
     TE DI FONDO UNA GRANDE     "
     ;
5910 PRINT"STATUA DEL DIO-ELEFAN
     TE. LA SUA PROBO-   ";
5920 PRINT"SCITE OSCILLA. SEMBRA
      CHIEDERE QUALCOSA."
5930 N=0:S=0:O=13:E=0:GOTO 390
5999 REM  LUOGO 15
6000 PRINT"[CLEAR]SEI USCITO DAL
      TEMPIO. LA PACE E LA      "
     ;
6010 PRINT"PROSPERITA' TI ACCOMP
     AGNERANNO."
6020 N=0:S=0:O=0:E=0:END
6099 REM  LUOGO 17
6100 PRINT"[CLEAR]UN LUNGO CORRI
     DOIO CON UNA STRETTA PAS- "
6110 PRINT"SATOIA."
6120 N=12:S=22:O=0:E=0:GOTO 390
6199 REM  LUOGO 22
6200 PRINT"[CLEAR]UNA PICCOLA ST
     ANZA CON LUMINOSI DISEGNI."
6210 N=17:S=0:O=0:E=23:GOTO 390
6299 REM  LUOGO 23
6300 PRINT"[CLEAR]UNA VASTA STAN
     ZA CON PREZIOSI DIPINTI.   "
6310 N=0:S=0:O=22:E=0:GOTO 390
```

Metti un tigre nel tuo 128

Una piccola incompatibilità del 64/128 da sfruttare a nostro vantaggio

Di Roberto Marigo

La stupenda tastiera del Commodore 128, e la velocità di 2 Mhz del suo microprocessore, sembrano purtroppo utilizzabili solo in modalità 128 e, nel secondo caso, addirittura se in possesso del monitor a 80 colonne.

Portando il computer in modalità C/64, tali caratteristiche vanno perdute, a meno di non utilizzare particolari accorgimenti (vedi il programma "Supertastiera" pubblicato su CCC N.35).

Nell'articolo citato, infatti, era possibile utilizzare il C/64 a 2Mhz, a patto però di rinunciare al video, nel senso che, ad alta velocità, lo schermo si riempiva di caratteri incomprensibili che scomparivano solo tornando alla velocità normale.

Il programma di queste pagine permette di ottenere contemporaneamente la tastiera "estesa" e la velocità di 2Mhz della CPU senza dover rinunciare al video. In questo caso l'aumento di velocità è di circa il 50% rispetto al normale funzionamento del Commodore 128 in modo 64.

L'utilità di una simile modifica è intuitiva: da un lato sveltire il processo di input di lunghe liste di dati attraverso il comodissimo tastierino apposito, e dall'altro aumentare la velocità di esecuzione di un programma.

Come usare il programma

Interamente in Linguaggio Macchina, il programma permette, tramite SYS, di attivare o disattivare il tastierino numerico e la nuova frequenza di clock.

Dopo il caricamento delle linee di data, il computer visualizzerà, con somma gioia degli smanettoni, gli "entry points" delle varie routine che utilizza:

SYS 49152 - tastiera espansa
SYS 49232 - tastiera normale
SYS 49440 - clock a 2Mhz
SYS 49556 - clock a 1Mhz

Due sono le routine del programma, utilizzabili contemporaneamente, che provvedono ad altrettanti compiti.

La prima modifica la routine di scansione della tastiera (keyboard scanning) affinchè possa leggere tutti i tasti. Per quanto riguarda i tasti di

---

Nel digitare righe di programma basic che contengono istruzioni DATA, è piuttosto facile incorrere in errori di digitazione. Supponiamo che un'ipotetica linea basic numerata con 1200 debba contenere i tre valori: 123, 456, 789. Ecco alcuni esempi di errori più frequentemente commessi:

1200 DATA,123,456,789

C'è una virgola dopo la parola "DATA". I dati letti dal computer sono, in questo caso, quattro: 0, 123, 456, 789. Se, infatti, non figura alcun carattere dopo l'istruzione DATA, automaticamente viene assunto il valore nullo (0).

1200 DATA 123,456,789,

In questo caso, dopo il numero 789, il computer, grazie alla presenza della virgola erroneamente inserita, "crede" che ci sia un altro valore e, non trovandolo, lo assume come nullo (0).

1200 DATA 1234,56,789

La virgola è posizionata male, vale a dire dopo il carattere "4" e non dopo il carattere "3". Il computer non può sapere se il valore esatto è 123 oppure 1234 e individuare un errore, in questo caso, risulta piuttosto laborioso.

ESC, TAB, ALT, HELP, LINE FEED, NO SCROLL, ad essi assegna i seguenti codici ASCII :

ESC normale: Chr$ (27)
ESC shiftato: Chr$ (27)

TAB normale: Chr$ (9)
TAB shiftato: Chr$ (24)

ALT normale: Chr$ (14)
ALT shiftato: Chr$ (142)

HELP normale: Chr$ (8)
HELP shiftato: Chr$ (9)

LINE FEED normale: Chr$ (10)
LINE FEED shiftato: Chr$ (10)

NO SCROLL normale: Chr$ (3)
NO SCROLL shifato: Chr$ (3)

La seconda routine cambia la frequenza di clock del sistema a 2 Mhz e modifica l'interrupt in modo tale da consentire la gestione anche delle 40 colonne: il risultato, essendo un compromesso, limita la percentuale di incremento della velocità al 50%

I programmi dimostrativi

Assieme al programma presentiamo due programmi utili per dimostrare l'incremento della velocità del computer.

Il primo è un semplice ciclo di ritardo con le istruzioni FOR-NEXT da 1 a 10000; il secondo genera 28 stringhe nell'array X$, ognuna di 5 caratteri casuali, e poi le ordina alfabeticamente con una procedura di Bubble Sort.

Limiti di utilizzo

Questo programma quindi è utilissimo in molte delle applicazioni di gestione di calcoli con il C/64. Unico suo neo è l'impossibilità per la CPU di colloquiare con le periferiche: pertanto si consiglia di disattivare la velocità di 2 Mhz quando si desideri registrare, caricare, stampare dati con una periferica.

Un altro limite è rappresentato dal fatto che, caricando programmi professionali, si rischia di veder bloccare il sistema oppure di ritornare alla velocità e tastiera "normali" dal momento che molti programmi, a causa di protezioni varie, modificano l'interrupt appena vengono caricati, e non c'è modo di evitare questo inconveniente.

```
100 REM SUPER-64
110 REM BY ROBERTO MARIGO
120 :
130 FOR I=49152 TO 49660:READ A
    :Y=Y+A:POKE I,A:NEXTI
140 IF Y<>59049 THEN PRINT"ERRO
    RE DI TRASCRIZIONE":END
150 POKE 53280,7:POKE 53281,6:P
    RINTCHR$(147)"SUPER 64":PRI
    NT:PRINT
160 PRINT TAB(5)"SYS 49152 :SUP
    ER-TASTIERA"

170 PRINT TAB(5)"SYS 49232 :TAS
    TIERA NORMALE":PRINT
180 PRINT TAB(5)"SYS 49440 :SUP
    ER VELOCITA'"
190 PRINT TAB(5)"SYS 49556 :VEL
    OCITA' NORMALE"
200 END

210 DATA 8,72,120,173,20,3,201,
    110,173,21,3,233,192,240,12
    ,173
220 DATA 20,3,141,232,192,173,2
    1,3,141,233,192,173,2,3,201
    ,74
230 DATA 173,3,3,233,192,240,12
    ,173,2,3,141,234,192,173,3,
    3
240 DATA 141,235,192,169,110,14
    1,20,3,169,192,141,21,3,169
    ,74,141
250 DATA 2,3,169,192,141,3,3,10
    4,40,96,44,0,192,108,234,19
    2
260 DATA 8,72,120,173,232,192,1
    41,20,3,173,233,192,141,21,
    3,173
270 DATA 234,192,141,2,3,173,23
    5,192,141,3,3,104,40,96,169
    ,64
280 DATA 133,203,41,0,141,47,20
    8,9,255,141,0,220,205,1,220
    ,240
290 DATA 66,160,89,169,251,141,
    47,208,72,173,1,220,205,1,2
    20,208
300 DATA 248,162,8,10,176,2,132
    ,203,136,192,65,144,7,202,2
    08,243
310 DATA 104,74,16,225,104,173,
    141,2,10,201,8,144,2,169,6,
    170
320 DATA 189,224,192,133,245,18
    9,225,192,133,246,32,213,19
    2,32,224,234
330 DATA 32,213,192,169,255,141
    ,47,208,41,127,141,0,220,16
    4,203,140
340 DATA 236,192,108,232,192,16
    5,197,174,236,192,141,236,1
    92,134,197,96
350 DATA 171,192,195,192,195,19
```

```
    2,171,192,49,234,131,164,64
    ,8,56,53
360 DATA 9,50,52,55,49,27,43,45
    ,10,13,54,57,51,14,48,46
370 DATA 145,17,157,29,3,9,56,5
    3,24,50,52,55,49,27,43,45
380 DATA 10,13,54,57,51,142,48,
    46,145,17,157,29,3,0,255,0
390 DATA 8,72,120,173,20,3,201,
    191,173,21,3,233,193,240,12
    ,173
400 DATA 20,3,141,248,193,173,2
    1,3,141,249,193,173,2,3,201
    ,142
410 DATA 173,3,3,233,193,240,12
    ,173,2,3,141,250,193,173,3,
    3
420 DATA 141,251,193,169,191,14
    1,20,3,169,193,141,21,3,169
    ,142,141
430 DATA 2,3,169,193,141,3,3,16
    9,0,141,48,208,173,18,208,2
    08
440 DATA 251,169,49,141,18,208,
    173,17,208,41,127,141,17,20
    8,169,1
450 DATA 141,48,208,173,26,208,
    9,1,141,26,208,104,40,96,44
    ,32
460 DATA 193,108,250,193,8,72,1
    20,173,26,208,41,254,141,26
    ,208,169
470 DATA 0,141,48,208,173,248,1
    93,141,20,3,173,249,193,141
    ,21,3
480 DATA 173,250,193,141,2,3,17
    3,251,193,141,3,3,104,40,96
    ,172
490 DATA 25,208,48,7,173,13,220
    ,88,108,248,193,152,74,144,
    249,140
500 DATA 25,208,173,48,208,73,1
    ,141,48,208,74,176,3,169,25
    0,44
510 DATA 169,49,141,18,208,104,
    168,186,189,4,1,208,3,222,5
    ,1
520 DATA 222,4,1,104,170,104,40
    ,96,110,192,74,192,0
530 END
```

```
100 REM DEMO PER SUPER-64
110 :
120 REM CARICARE E LANCIARE IL
    PROGRAMMA IN L.M. PRIMA DI
    UTILIZZARE LA PRESENTE
130 REM ROUTINE DIMOSTRATIVA
140 :
150 SYS49556:PRINTCHR$(147); TA
    B(10)"ATTENDI 29...":UU=-1
160 DIM X$(29)
170 FOR I=0 TO 29:REM GENERA CA
    SUALMENTE 100 STRINGHE DA 5
     CARATTERI CIASCUNA E
180 FOR II=0 TO 4:REM METTILE N
    EL VETTORE X$ DA 0 A 59
190 X$(I)=X$(I)+CHR$(RND(.)*26+
    65)
200 PRINTCHR$(19)I:NEXTII:NEXTI
    :REM  ESEGUI UN SORT ALFABE
    TICO DELLE STRINGHE
210 TI$="000000"
220 F=0:FOR I=0 TO 28:PRINTCHR$
    (19)CHR$(17)I
230 IF X$(I)<X$(I+1) THEN X$=X$
    (I):X$(I)=X$(I+1):X$(I+1)=X
    $:X$="":F=1
240 NEXTI
250 IF F THEN 220
260 UU=UU+1:IF UU=1 THEN PRINT:
    PRINT"SECONDO ";
270 PRINT"TEMPO IMPIEGATO:"TI$
280 SYS49440:PRINT:IF UU=0 THEN
     CLR :GOTO 160
290 SYS49232:SYS49556:REM RIPRI
    STINO VALORI NORMALI
```

```
100 REM DEMO 1 PER SUPER 64
110 :
120 SYS49556
130 TI$="000000"
140 FOR I=0 TO 10000:NEXT
150 PRINT"TEMPO IMPIEGATO NORMA
    LMENTE :"TI$
160 SYS49440
170 TI$="000000".
180 FOR I=0 TO 10000:NEXT
190 PRINT"TEMPO IMPIEGATO CON S
    UPER-64:"TI$
200 SYS49556
```

Mini disegni con la tua stampante

Un simpatico programma di Hard Copy di schermate in bassa risoluzione per le applicazioni più disparate

di Gabriele Casetti

Questo brevissimo programma consente di disegnare sullo schermo del C/64, in bassa risoluzione, una qualsiasi figura e, in seguito, di riportarla su stampante.

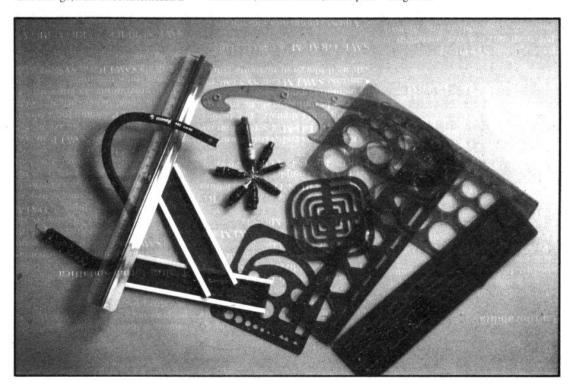
Ciò che viene riprodotta è una piccola immagine larga 40 dot (=pixel di stampante) e alta 25 dot.

Come avrete intuito guardando le figure pubblicate, ogni carattere dello schermo sarà rappresentato da un punto nero su carta, mentre lo spazio bianco non verrà tracciato.

La tecnica di programmazione adoperata sarà preziosissima per coloro che desiderano approfondire le proprie conoscenze sull'alta risoluzione della stampante, dal momento che si sfrutta intensamente la programmazione dei caratteri.

Il programma è suscettibile di notevoli migliorie, come la possibilità di memorizzare, su nastro o disco, i disegni eseguiti.

Il tempo di elaborazione può sembrare lungo, a causa della lentezza tipica del Basic. Chi possiede un compilatore potrà ovviare all'inconveniente senza alcuna difficoltà.

Al lettore, come al solito, il compito di effettuare eventuali personalizzazioni e di... divertirsi con questo listato-gadget certamente nuovo e originale.

```
100 REM MINI-SCHERMATE SU STAMP
    ANTE
110 REM PER C/64 & STAMPANTI MP
    S-803 E COMPATIBILI
120 REM BY GABRIELE CASETTI
130 REM FIRENZE
140 :
150 DIM A%(39):POKE 650,128
160 GOSUB 410
170 FOR T=50151 TO 50311:POKE T
    ,32:NEXT
180 POKE 53280,0:POKE 53281,2:P
    OKE 646,0
190 POKE 198,2:POKE 631,20:POKE
     632,20
200 INPUT "[CLEAR]";A$
210 FOR T=0 TO 999:POKE 49152+T
    ,PEEK(1024+T):NEXT
220 INPUT "[CLEAR][NERO]QUANTE
    NE STAMPO MAX 12";N
230 IF N<1 OR N>12 THEN 220
240 INPUT "[DOWN]MARGINE SIN. I
    N DOT";M
250 MM=480-N*40
260 IF M<0 OR M>MM THEN PRINTMM
    "MAX":GOTO 240
270 OPEN 1,4:PRINT#1,CHR$(13)CH
    R$(8);
280 FOR RG=0 TO 3
290 PRINT#1,CHR$(15) TAB(M/6)CH
    R$(8);
300 FOR X=0 TO 39:G=128
310 FOR R=0 TO 6:I=49152+X+RG*7
    *40+R*40
320 IF PEEK(I)<>32 AND PEEK(I)<
    >96 THEN G=G+2↑R
330 NEXT:A%(X)=G:NEXT
340 FOR C=1 TO N:FOR X=0 TO 39:
    PRINT#1,CHR$(A%(X));:NEXT:N
    EXT
350 PRINT#1:NEXT
360 PRINT#1,CHR$(15)CHR$(13)CHR
    $(13):CLOSE 1
370 PRINT"[CLEAR]ANCORA (S/N) ?
380 GET A$:IF A$="S" THEN 220
390 IF A$<>"N" THEN 380
400 END
410 POKE 53280,6:POKE 53281,4:P
    OKE 646,0
420 PRINT"[CLEAR]    MINI-DISEG
    NINI CON LA STAMPANTE"
430 PRINT"
    BY DINOSOFT"
440 PRINT"[DOWN]PREMI UN TASTO
    E VEDRAI LO SCHERMO PULI TO
    ."
450 PRINT"[DOWN]MUOVI IL CURSOR
    E CON GLI APPOSITI TASTI"
460 PRINT"E FAI IL TUO DISEGNO
    TENENDO PRESENTE  CHE:"
470 PRINT"[DOWN]SPAZIO = PUNTO
    NON STAMPATO
480 PRINT"[DOWN]GLI ALTRI CARAT
    TERI = [RVS] PUNTO STAMPATO
     [RVOFF]"
490 PRINT"[DOWN] ALLA FINE PREM
    I RETURN"
500 POKE 198,0:WAIT 198,1
510 RETURN
```

SORRIDI! SORRIDI! SORRIDI! SORRIDI! SORRIDI! SORRIDI! SORRIDI! SORRIDI! SORRIDI! SORRIDI!

Il primo giorno di scuola

Le prime applicazioni grafiche del nuovissimo simulatore di Gw-Basic

di Giancarlo Mariani e Fabio Sorgato

Il Gw-Basic, come tutti sapranno, è un linguaggio interprete che gira su computer Ms-Dos, vale a dire IBM e compatibili.

Come annunciato il mese scorso è ora disponibile in edicola una versione anche per il Commodore 64 e la presente rubrica è dedicata al "nuovo" linguaggio.

Lo scopo principale di queste pagine, dedicate agli acquirenti della nostra cassetta (e agli utenti dei Commodore PC 10, 20 e compatibili), sarà quello di illustrare in dettaglio le istruzioni di programmazione e, soprattutto, mettere in evidenza le similitudini e diseguaglianze tra la versione originale e la nostra.

La caratteristica principale del Gw-Basic è quella di essere molto versatile, ovvero di fornire un set di istruzioni idoneo a risolvere facilmente i molteplici problemi che sorgono in fase di strutturazione di un programma.

La versione C/64

Considerata la notevole diffusione dello standard IBM e compatibili, e le numerose richieste pervenute da parte dei lettori, abbiamo pensato di fare cosa gradita realizzando una versione del Gw-Basic funzionate anche sul Commodore 64.

Questa, ovviamente, non ha la pretesa di essere uno strumento di programmazione essendo in certi punti molto più "ristretta" e meno potente del corrispondente Basic IBM, ma, scritta principalmente a scopo didattico, ha il notevole pregio di avvicinare l'utente del piccolo Commodore al "misterioso" mondo dei personal computer.

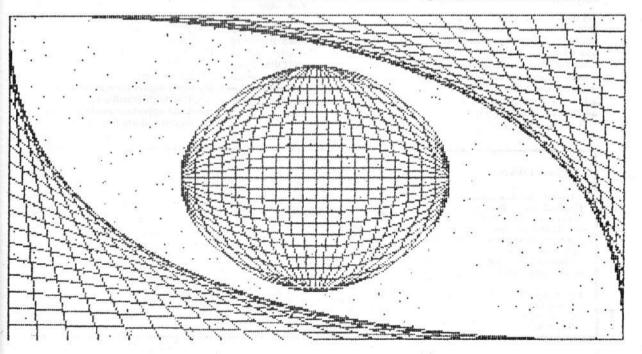
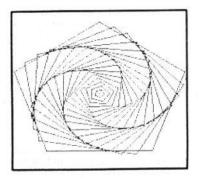
Dal momento che la maggior parte delle scuole di informatica (e non) ha adottato il PC-IBM come strumento didattico, la nostra realizzazione offre la possibilità agli studenti, che non possiedono un "compatibile", di esercitarsi anche a casa, risparmiando quindi tempo e riuscendo ad ottenere risultati migliori che non lavorando soltanto a scuola.

I listati Gw-Basic che pubblichiamo, e pubblicheremo, saranno in grado di funzionare allo stesso modo sul C/64 e su un Ms-Dos senza apportare modifiche; altri listati, invece, richiederanno lievi variazioni, dovute soprattutto alle differenze intrinseche tra le due macchine, costruite originalmente su standard (e microprocessori) completamente diversi.

La rubrica affronterà principalmente due temi diversi: il primo proporrà listati funzionanti indifferentemente su entrambe le macchine allo scopo di facilitare la comprensione delle istruzioni del Basic; il secondo tratterà dettagliatamente ogni singola istruzione complementando il manuale di istruzioni fornito con il nastro cassetta, e rivelerà anche alcuni "trucchi" atti ad utilizzare al meglio le caratteristiche del linguaggio.

Il primo impatto con il gw-Basic

Dopo avere caricato il Gw-Basic apparirà la schermata iniziale e saranno così disponibili un centinaio di comandi aggiuntivi oltre a quelli standard Commodore.

La prima differenza che balza all'occhio, a parte il diverso numero di byte liberi e la diversa intestazione, è

la presenza di messaggi visualizzati sulle ultime due righe di schermo.

Queste non sono altro che le intestazioni di quattro degli otto tasti funzione programmabili e non saranno in alcun modo alterate da eventuali sovrascritture o scroll di schermo.

Se risultassero fastidiose per qualche particolare applicazione sarà sufficiente disabilitarle con il comando KEYOFF (KEYON per riattivarle). Ricordiamo che il comando analogo per il "vero" Gw-Basic richiede lo spazio tra le due parole (Key Off).

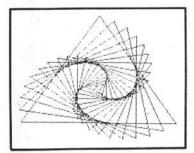
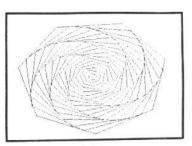
A questo punto si è pronti per scrivere programmi servendosi delle nuove isruzioni. Ribadiamo ancora una volta che TUTTE le "vecchie" istruzioni del Basic Commodore sono ancora disponibili: ciò permette una totale compatibilità con i listati Basic già in vostro possesso, compresi quelli che gestiscono gli sprite! (a patto di rinunciare al comando Keyon).

Un semplice programma

Per prendere familiarità con il nuovo sistema provate a digitare il seguente programmino:

```
10 SCREEN 1:CLS 1
20 CIRCLE (160,100),80
30 LINE (50,10)-(260,180),1,B
40 PAINT(160,100)
50 IF INKEY$<>CHR$(32)THEN 50
60 SCREEN 0:CLS
```

Dopo il RUN comparirà una circonferenza racchiusa in un rettangolo;, per tornare al modo testo è suffi-

ciente premere la barra spaziatrice. Il programma è autoesplicativo per cui lasciamo il compito al lettore di individuare il funzionamento delle varie istruzioni.

In modo Gw-Basic i tasti Run/Stop e Restore sono disabilitati; il primo è sostituito dalla combinazione di tasti CTRL-C mentre il RESTORE, da solo, va usato soltanto in caso di necessità poichè provoca non solo un restart del Basic, ma pone lo schermo in modo testo e non salva i puntatori per l'eventuale CONT.

Sarà bene sottolineare che l'istruzione Paint (che colora un'area delimitata di schermo) richiede, in certi casi, un tempo relativamente lungo. Durante il suo funzionamento la tastiera viene disabilitata e la pressione del tasto RESTORE sembra non sortire alcun effetto. Sarà quindi sufficiente attendere che il comando venga completamente eseguito prima di "riprendere" il programma.

I primi listati

In questa prima "puntata" abbiamo deciso di inserire alcuni listati funzionanti senza variazioni sia su IBM (dotato di scheda grafica) che su Commodore dotato di Gw-Basic.

Sono tutti semplici dimostrativi di tipo grafico, ciascuno dotato di particolari funzioni del Basic e non richiedono alcun commento grazie alla loro semplicità.

Il lettore inesperto è naturalmente invitato a digitarli cercando di capirne il funzionamento ed eventualmente ad apportare modifiche e variazioni di suo gradimento.

Anche su disco!

Coloro che desiderano procurarsi Gw-Basic su disco devono inviare in Redazione la modica cifra di L.25000, (oltre a L.3000 per le spese di spedizione). Non ci è possibile, infatti, inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo, ma anche il nome del software desiderato:

"Simualtore di Gw-Basic" su disco.

Chi volesse ricevere più celermente la confezione può inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale.

Un normale modulo di conto corrente, infatti, impiega circa due mesi (provare per credere!) per giungere al nostro indirizzo. Una lettera, invece, arriva anche in una sola settimana.

Inviate l'assegno, accompagnato da una lettera in cui specificate la richiesta, al solito indirizzo:

Systems Editoriale

Viale Famagosta, 75

20142 Milano

```
100 REM CRAZY BAR
110 REM VERSIONE MS-DOS
120 REM CBM 64 + GW-BASIC SIMULATOR
130 :
140 SCREEN 1:CLS:CLS 1
150 X1=50:Y1=50:X2=150:Y2=100
160 XI=5:YI=5:XJ=5:YJ=5
170 LINE (XA,YA)-(XB,YB),0
180 LINE (X1,Y1)-(X2,Y2),1
190 XA=X1:YA=Y1:XB=X2:YB=Y2
200 IF X1+XI<0 OR X1+XI>320 THEN XI=-XI
210 IF X2+XJ<0 OR X2+XJ>320 THEN XJ=-XJ
220 IF Y1+YI<0 OR Y1+YI>200 THEN YI=-YI
230 IF Y2+YJ<0 OR Y2+YJ>200 THEN YJ=-YJ
240 X1=X1+XI:X2=X2+XJ
250 Y1=Y1+YI:Y2=Y2+YJ
260 GOTO 170
```

```
10 REM POLIGONI
20 :
30 CLS
40 PI=3.14159265
50 INPUT "QUANTI LATI";L
60 SCREEN 1:CLS 1
70 PSET (160,20)
80 FOR K=-PI TO PI STEP 2*PI/L
90 LINE -(SIN(K)*85+160,COS(K)*80+100)
100 NEXT K
110 LINE -(160,20)
120 IF INKEY$<>" " THEN 120
130 SCREEN 0:RUN
```

```
5 REM STAR
6 :
10 SCREEN 1:CLS 1:CLS
15 PI=3.14159265
20 FOR K=0 TO 200 STEP 10
30 LINE (0,K)-(K*1.6,200),1
40 LINE (K*1.6,0)-(320,K),1
50 NEXT K
60 FOR K=.00001 TO 1 STEP .1
70 CIRCLE (160,100),70,1,0,2*PI,K
75 CIRCLE (160,100),70*K,1,0,2*PI,K
80 NEXT K
90 FOR K=1 TO 500
100 X=RND(1)*320:Y=RND(1)*200
110 PSET (X,Y)
120 NEXT K
130 IF INKEY$<>CHR$(13) THEN 130
```

OROLOGIO

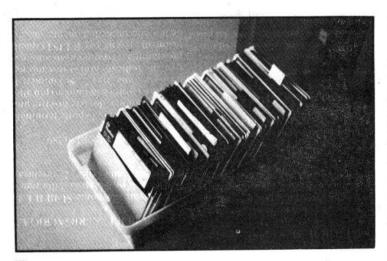
```
10 REM ROTAZIONI
20 :
30 CLS
40 PI=3.14159265
50 INPUT "QUANTI LATI";L:IF L<1 THEN 30
60 SCREEN 1:CLS 1
70 PSET (160,100)
80 FOR J=0 TO 90 STEP 5:A=0
90 FOR K=(PI/90)*J-PI TO PI+(PI/90)*J+PI/10 STEP 2*PI/L
100 IF A=0 THEN A=1:PSET (SIN(K)*(J+J/16)+160,COS(K)*J+100)
110 LINE -(SIN(K)*(J+J/16)+160,COS(K)*J+100)
120 NEXT K:NEXT J
130 IF INKEY$<>" " THEN 130
140 SCREEN 0:RUN
```

POLIGONI

```
5 REM TIME
6 :
10 CLS:PI=3.14159265
20 DIM AX(60),AY(60),BX(60),BY(60),CX(60),CY(60)
30 DIM DX(60),DY(60)
40 A=0:FORK=PI TO -PI STEP -PI/30
50 X=SIN(K):Y=COS(K)
60 AX(A)=X*90+160:AY(A)=Y*90+100
70 BX(A)=X*80+160:BY(A)=Y*80+100
80 CX(A)=X*70+160:CY(A)=Y*70+100
90 DX(A)=X*50+160:DY(A)=Y*50+100
100 A=A+1:LOCATE 1,1:PRINT A:NEXT
110 CLS:PRINT"INSERISCI L'ORA (HH:MM:SS)"
120 LINEINPUT B$:GOSUB 1000:IF LEN(TIME$)>7 THEN TIME$=B$:ELSE TIME$=A$
130 SCREEN 1
140 CLS 1:CLS
150 FOR K=0 TO 59
160 PSET (AX(K),AY(K))
170 IF K/5<>INT(K/5) THEN 190
180 LINE (AX(K),AY(K))-(BX(K),BY(K)),1
190 NEXT K
200 B$=TIME$:IF LEN(TIME$)>7 THEN GOSUB 1000:ELSE A$=TIME$
205 FOR K=1 TO 6 STEP 2
210 A(INT(K/2))=VAL(MID$(A$,K,2)):NEXT K:A(0)=12*(A(0)/12-INT(A(0)/12))
220 LINE (160,100)-(CX(A(2)),CY(A(2))),1
230 IF A1<>A(2) THEN LINE (160,100)-(CX(A1),CY(A1)),0:A1=A(2)
240 LINE (160,100)-(CX(A(1)),CY(A(1))),1
250 IF A2<>A(1) THEN LINE (160,100)-(CX(A2),CY(A2)),0:A2=A(1)
260 L=A(0)*5+INT(A(1)/12):LINE (160,100)-(DX(L),DY(L)),1
270 IF L<>A3 THEN LINE (160,100)-(DX(A3),DY(A3)),0:A3=L
279 C$=A$
280 B$=TIME$:IF LEN(TIME$)>7 THEN GOSUB 1000:ELSE A$=B$
285 IF A$=C$ THEN 280
290 GOTO 200
1000 A$=LEFT$(B$,2)+MID$(B$,4,2)+RIGHT$(B$,2):RETURN
```

QUALSIASI COMPUTER

Directory N.4

Che cosa c'è nella nuova Directory?

I nostri lettori sono ormai abituati alle sorprese (in senso buono...) e, ormai, il fatto che vi siano tanti file, da riempire quasi per intero il disco, non fa più notizia.

Stavolta ricorderemo soltanto che su Directory vengono inseriti TUTTI i programmi pubblicati su questo fascicolo di Commodore Computer Club, TUTTI i programmi Commodore che è possibile trovare sulla rivista Personal Computer che, in contemporanea, potete trovare in edicola, e altri file vari che, almeno finora, sono stati apprezzati in pieno dai nostri lettori: musiche nell'interrupt da usare nei vostri listati, set di caratteri ridefiniti, utility varie e così via.

Non diremo con precisione il contenuto del dischetto, anche perchè abbiamo deciso, fin dal numero scorso, di aggiungere all'ultimo momento file e programmi che riteniamo interessanti.

Ma c'è di più: per accontentare i lettori che hanno "fretta", e che non intendono aspettare i (lunghi) tempi del servizio postale, è possibile ordinare, oltre alle copie arretrate di Directory, anche quella... prossima.

Nel momento, infatti, in cui state leggendo queste pagine, in Redazione abbiamo già deciso che cosa pubblicare sul prossimo fascicolo (sia di C.C.C. sia di Personal Computer) e stiamo per mettere a punto il dischetto N.5.

Chi lo desidera, pertanto, può ordinare fin d'ora DIRECTORY N.5 dal momento che, nel tempo che la vostra lettera contenente l'assegno arriva a destinazione, avremo a disposizione le copie del floppy relative a CCC N.39.

Naturalmente le spese di spedizione (L.3000) vanno conteggiate una sola volta, da aggiungere al prezzo di L.12000 per ciascun dischetto, della serie Directory, richiesto.

Partecipazione dei lettori

Centinaia sono le occasioni per partecipare a "Directory".

Prima di inviare il risultato del vostro lavoro vi consigliamo, però, di telefonarci per stabilire se risponde ai requisiti per l'eventuale pubblicazione (tel. 02/84.67.34.8)

Come procurarsi "Directory"

Avvertiamo i lettori che NON è assolutamente possibile inviare i programmi su nastro, per intuibili motivi di economia ed affidabilità del nastro cassetta.

Ogni numero di "Directory" può quindi esser richiesto SOLO su disco inviando L.12000 per ciascun disco oltre a L.3000 (fisse) per le spese di spedizione (indipendenti dal numero di dischi richiesti).

Non ci è possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo, ma anche il nome del disco desiderato:

"Directory N.4"

N.B. Per ottenere il materiale ordinato in tempi più ristretti, inviate l'importo a mezzo assegno bancario non trasferibile: le poste italiane non brillano per velocità! (due mesi circa per il recapito di un C/C postale).

Tre nuovi comandi

Ecco a voi tre routine L.M. utili anche a chi programma soltanto in Basic

di Armando Sforzi

Comando Instr (22711/22781)

Capita sovente di dover controllare se all'interno di una stringa alfanumerica sia presente una particolare successione di caratteri, in modo da sostituirli, cancellarli, od altro.

Il Basic del Commodore 64 non prevede un comando specifico e ne consegue che tale operazione va attuata tramite altre istruzioni, con inevitabile complicazione del programma.

Ad esempio, in Basic V2, il problema esposto potrebbe essere risolto nel seguente modo:

```
10 A$="Roma,Milano,Venezia"
15 B$="Milano":P=0
20 FOR J=1 TO LEN(A$)-LEN(B$)
30 IF MID$(A$,J,LEN(B$))=B$ THEN
P=J:J=LEN(A$)
40 NEXT
```

Con la routine L.M. pubblicata, lunga 71 byte, che simula l'istruzione INSTR dei Basic più evoluti, il problema presentato nell'esempio si risolve scrivendo:

SYS XXXXX A$,B$

Nella sintassi dovranno comparire solo variabili e non direttamente le stringhe.

Il risultato dell'operazione è contenuto nel byte 781 (immagine del registro X del 6510). Se questo vale 0 significa che B$ non è stata trovata in A$.

Altri valori indicano, invece, la posizione di partenza di B$ in A$.

```
1000 PRINTCHR$(147)"QUESTA ROUNT
     INE LA CUI SINTASSI E':
      SYSXXXXX V1$,V2$"
1010 PRINT"RIPORTA NEL BYTE 781
     LA POSIZIONE DI     PARTENZA
     "
1020 PRINT"DELLA STRINGA V2$ CON
     TENUTA IN V1$        SE V2$ N
     ON SI TROVA IN V1$"
1030 PRINT"IL CONTENUTO DEL BYTE
      781 SARA' 0"
1040 RETURN
1050 DATA 32,139,176,160,0,177,7
     1,133,182,200,177,71,133,25
     1,200
1060 DATA 177,71,133,252,32,253,
     174,32,139,176,160,0,177,71
1070 DATA 133,189,200,177,71,133
     ,253,200,177,71,133,254
1080 DATA 162,1,160,0,177,251,20
     9,253,240,14,230,251,208,2,
     230,252
1090 DATA 232,198,182,208,237,16
     2,0,96,200,196,189,144,231,
     96,-1,10951
```

```
  0 REM DEMO
  1 REM SIMULA COMANDO INSTR
  2 REM ESTRAE CASUALMENTE I PR
    IMI 40 NUMERI SENZA RIPETIZ
    IONE
  3 REM INIZIO LM=22711
  4 REM BY ARMANDO SFORZI
  5 :
100 J=0:TI$="000000"
110 N1$="COMMODORE COMPUTER CLU
    B"
111 N2$="SYSTEMS EDITORIALE"
112 N3$="COMPUTER"
115 SYS22711N1$,N3$
120 IF PEEK(781) THEN PRINT"IN
    N1$ E' PRESENTE "N3$" A PAR
    TIRE DA"PEEK(781)
125 SYS22711N2$,N3$
127 IF PEEK(781) THEN PRINTPEEK
    (781)
```

Come utilizzare le routine

Sul N.31 di Commodore Computer Club è iniziata una nuova rubrica che ha lo scopo di venire incontro ai principianti (senza trascurare gli esperti), che desiderano potenziare al massimo le caratteristiche del proprio computer.

Il Basic presenta, infatti, carenze notevoli che possono esser limitate ricorrendo all'uso di routine in linguaggio macchina (LM): è sufficiente attenersi alle istruzioni pubblicate per utilizzare i sottoprogrammi LM con la massima semplicità. Gli "esperti" potranno fare a meno di seguire le istruzioni ed utilizzare direttamente i programmi L.M. pubblicati.

I principianti, invece, è opportuno che leggano con attenzione le "istruzioni per l'uso".

0' Se questa è la prima volta che leggete la rivista, accendete il vostro Commodore 64 e saltate al punto N.2.

1' Accendete il computer e, se desiderate "fondere" alcune (o tutte) le routine di questo numero con quelle tratte dai numeri precedenti (a patto, ovviamente che ne siate in possesso), caricate il file-programma "Nuovo Sistema" (nome standard adottato) con una delle due forme sintattiche che si riferiscono, rispettivamente, ai possessori di nastro o disco:

Load "Nuovo Sistema",1,1
Load "Nuovo Sistema",8,1

Subito dopo digitate NEW e premete il tasto Return.

2' Caricate il programma "Fissa Top di memoria" e lanciatelo col solito RUN. Alla domanda "Ultima locazione?" digitate 20000 e, alla successiva richiesta di conferma, premete il tasto "S". Le altre informazioni che appaiono sul video possono esser comprese solo dagli esperti: i principianti possono tranquillamente ignorarle e saltare alla prossima fase (N.3).

3' Caricate (o digitate dalla rivista) il programma "Caricatore".

4' Digitate dalla rivista la routine che interessa (scritta sempre in Basic, contenente in prevalenza istruzioni Data e numerata da 1000 in poi).

5' Effettuate una copia di sicurezza del programma che rappresenta la "fusione" dei due listati ("Caricatore" + routine Basic pubblicata).

6' Dopo aver digitato Run, alla domanda "Da quale locazione?" rispondete con l'indirizzo iniziale suggerito nello stesso titolo della routine in oggetto. Se il computer, dopo alcuni secondi, visualizza, come indirizzo finale, un valore diverso da quello pubblicato nel titolo (oppure il messaggio "Errore di trascrizione"), interrompete il lavoro (tasti Run Stop e Restore) e verificate con attenzione quanto avete trascritto da rivista.

Se, invece, compaiono messaggi "confortanti" (Routine allocata da... a... Attivare con Sys... ed altre informazioni comprensibili dagli esperti), digitate il programma dimostrativo e lanciatelo: da questo momento avete a disposizione una nuova routine LM da attivare mediante SYS come indicato nelle istruzioni pubblicate per ciascuna routine.

7' Ripetete le operazioni, dal punto 3 in poi, per ciascuna routine pubblicata che intendete

FISSA TOP MEMO

```
100 PRINTCHR$(147)"FISSA TOP DI
     MEMORIA"
110 INPUT "ULTIMA LOCAZIONE";X:
    X=X-1
112 PRINT:PRINT"I VALORI ATTUAL
    I SONO:":PRINT
113 X1=INT(X/256):X2=X-(X1*256)
115 PRINT"PEEK(55):"PEEK(55):PR
    INT"PEEK(56):"PEEK(56)
116 PRINT"FRE(0):"FRE(0)
117 PRINT:PRINT"I VALORI NUOVI
    SAREBBERO:":PRINT:PRINT"PEE
    K(55):"X2
118 PRINT"PEEK(56):"X1
120 PRINT:PRINT"CONFERMI? (S/N)
    "
130 IF PEEK(197)=64 THEN 130
135 IF PEEK(197)<>13 THEN POKE
    198,0: RUN
150 POKE 55,X2:POKE 56,X1: RUN1
    60
160 PRINT"FRE(0):"FRE(0):PRINT:
    PRINT"NEW"
```

SAVE ZONA RAM

```
150 PRINTCHR$(147):INPUT "LOCAZ
    IONE INIZIALE";X
160 INPUT "LOCAZIONE FINALE";T
170 PRINT:PRINT"INIZIO:"X:PRINT
    "FINE:"T
180 IF T<=X THEN  RUN
190 PRINT:PRINT"CONFERMI? (S/N)
    "
200 GET A$:IF A$="" THEN 200
210 IF A$<>"S" THEN  RUN
220 PRINT:INPUT "NOME FILE";A$
230 PRINT"1- CASSETTA":PRINT"2-
     DISCO"
240 GET B$:IF B$="" THEN 240
250 IF B$="1" THEN W=1:GOTO 280
260 IF B$="2" THEN W=8:GOTO 280
270 GOTO 240
280 PRINTCHR$(147);
290 POKE 198,5:POKE 631,19:POKE
     632,13:POKE 633,13:POKE 63
    4,13:POKE 635,0
300 X1=INT(X/256):X2=X-(X1*256)
    :PRINT"P⌐44,"X1":P⌐43,"X2;
310 Y1=INT(T/256):Y2=T-(Y1*256)
    :PRINT":P⌐46,"Y1":P⌐45,"Y2
320 PRINT:PRINT:PRINT"S♠"CHR$(3
    4)A$CHR$(34)","W",1"
330 PRINT:PRINT:PRINT:PRINT:PRI
    NT"SYS64738"
```

CARICATORE

```
150 REM PER UTILIZZARLO, LEGGI
    LE ISTRUZIONI PUBBLICATE SU
160 REM COMMODORE COMPUTER CLUB
170 :
180 Y=-1:GOSUB 1000:PRINT:INPUT
     "DA QUALE LOCAZIONE";X
190 READ W:Y=Y+1:IF W<0 THEN 21
    0
200 GOTO 190
210 PRINT"PRIMA LOCAZIONE ="X
220 PRINT"ULTIMA LOCAZIONE ="X+
    Y-1:PRINT
230 PRINT"CONFERMI? (S/N)"
240 GET A$:IF A$="" THEN 240
250 IF A$="S" THEN RESTORE :T=X
    :GOTO 280
260  RUN
270 :
280 GOSUB 1000:PRINT:PRINT"ATTE
    NDERE...":PRINT:W=0
290 READ B:IF B>=0 THEN POKE T,
    B:T=T+1:W=W+B:GOTO 290
300 READ B:IF B<>W THEN PRINT:P
    RINTCHR$(18)"ERRORE DI TRAS
    CRIZIONE":END
310 PRINT"ROUTINE ALLOCATA DA"X
    "A"T-1"COMPR.":PRINT
320 PRINT"ATTIVARE CON SYS"X:PR
    INT
330 X1=INT(X/256):X2=X-(X1*256)
    :PRINT"POKE44,"X1":POKE43,"
    X2;
340 Y1=INT(T/256):Y2=T-(Y1*256)
    :PRINT":POKE46,"Y1":POKE45,
    "Y2:END
```

"collezionare" non dimenticando di digitare NEW dopo ogni felice conclusione della fase N.6. Ai principianti consigliamo vivamente di trascriverle tutte in modo da aumentare la propria esperienza e, soprattutto, per evitare incomprensioni degli articoli che leggeranno su Commodore Computer Club.

8 'Caricate, dopo un nuovo NEW, il programma "Save Zona Ram" e, dopo il Run, alla domanda "Da quale locazione?" rispondete con 20000. Alla seconda domanda "A quale locazione?" ripondete digitando l'indirizzo finale dell'ultima routine trascritta. A seconda se avete un registratore oppure un drive, sul nastro (oppure sul disco) vi ritroverete, dopo aver risposto alle varie domande, il file-programma "Nuovo Sistema" (nome che suggeriamo di assegnare quando compare la relativa domanda). Tale file-programma (da caricare come indicato al punto 1) sarà utilissimo sia per arricchire la vostra raccolta (trascrivendo le routine dei prossimi numeri di Commodore Computer Club), sia per utilizzarle in vostri listati.

9 'Digitate SYS 64738 oppure premete il tasto di Reset (se lo possedete) in modo da rimettere "a posto" il computer. Caricate il programma "Fissa Top di memoria" e rispondete con 20000 alla domanda che vi porrà: da questo momento potete disporre sia delle consuete istruzioni Basic che delle routine LM richiamabili con le corrispondenti SYS. Non dimenticate di ripetere la presente fase (N.9) tutte le volte che premete il tasto di Reset o dopo un reset software (SYS 64738). Se, invece, spegnete il computer, sarà necessario attuare la fase N.1 e N.2 per inserire nuovamente nel calcolatore le nuove routine ed usarle senza pericolo.

Collaborazione dei lettori

I lettori che intendono collaborare devono inviare (almeno) tre routine, relativi listati dimostrativi ed articoli esplicativi. Le norme da seguire per la stesura dei listati (piuttosto rigide, per ovvi motivi di compatibilità) sono state segnalate sul N.31. Per ulteriori informazioni, comunque, è possibile telefonare in Redazione (02/8467348) chiedendo di Michele Maggi.

Comando Mid$ (22599/22710)

Un'istruzione molto comoda, implementata sui Basic più evoluti di quello presente sul C/64, è MID$ usata non come funzione ma come comando, cioè posta a sinistra del segno di uguale.

Usata nel modo descritto permette di assegnare alcuni caratteri, appartenenti ad una stringa, ad un'altra stringa, a partire da una posizione specificata.

Per fare un esempio, banale ma chiarificatore, se avessimo la stringa:

A$="Alberto,Maria,Carlo"

e volessimo sostituire il nome "Maria" con "Lucia", dovremmo scrivere:

10 A$="Alberto,Maria,Carlo"
15 B$="Lucia":P=8
20 A$=LEFT$(A$,P)+B$+MID$(A$, P+LEN(B$)+1)

E' evidente come questa operazione sia macchinosa e piuttosto difficile da gestire.

Con la nostra routine (che occupa 112 byte) è possibile eseguire la stessa operazione con grande semplicità, scrivendo:

SYS XXXXX A$,P,B$

La sintassi del comando non permette l'uso diretto delle stringhe, che dovranno quindi comparire sotto forma di variabili. Si noti, inoltre, l'assenza della virgola (,) tra l'indirizzo della SYS (XXXXX) e la prima variabile (A$).

La routine, inoltre, controlla che il parametro che indica la posizione di partenza (P, nell'esempio) sia "legale", cioè maggiore di zero e minore o eguale alla lunghezza della stringa principale (A$).

Avremo segnalazione di "Illegal Quantity" nel caso in cui la stringa da inserire (B$), considerato il punto di partenza, sia troppo lunga e oltrepassi i limiti della prima stringa.

```
1000 PRINTCHR$(147)"QUESTA ROUTINE,LA CUI SINTASSI E'
     :"
1005 PRINT"SYSXXXXX V1$,P,V2$"
1010 PRINT"PERMETTE DI INSERIRE LA STRINGA V2$ DIRETT
     AMENTE IN V1$"
1020 PRINT"A PARTIRE DALLA POSIZIONE P"
1030 RETURN
1040 DATA32,139,176,160,1,177,71,133,251,200,177,71,1
     33,252
1050 DATA32,253,174,32,138,173,32,247,183,132,182,160
     ,0
1060 DATA177,71,133,189,208,5,162,11,108,0,3,197,182,
     176,5,162,14,108,0,3
1070 DATA32,253,174,32,139,176,160,1,177,71,133,253,2
     00,177,71,133,254
1080 DATA160,0,177,71,24,101,182,168,136,196,189,240,
     7,144,5,162,14,108,0,3
1090 DATA164,182,136,240,9,230,251,208,2,230,252,136,
     208,247,177,71
1100 DATA168,136,177,253,145,251,136,192,255,208,247,
     96,-1,15155
```

```
  0 REMDEMO
  1 REMSIMULA COMANDO MID$
  2 REMINIZIO LM=22599
100 A$=".............":REM GUARDA BENE QUESTA RIGA P
    RIMA DEL RUN
105 J=1:PRINT"DIGITA UN NUMERO:":PRINTTAB(18)CHR$(14
    5);
110 IFJ<1THENJ=1
115 GETN$:IFN$=""THEN110
117 PRINTN$;:IFN$=CHR$(20)ORN$=CHR$(157)THENJ=J-1:
    GOTO110
120 IFN$<>CHR$(13)THENSYS22599A$,J,N$:J=J+1:GOTO110
130 PRINT"VALORE[ 3 SPAZIO]IMMESSO:"VAL(A$)
140 PRINT"SORPRESA....":LIST100
150 END
```

Help Variabili (22782/22934)

Di norma, durante la messa a punto di un programma Basic, vengono inseriti comandi di Stop nei punti strategici del listato stesso.

Ad ogni Break è possibile, in tal modo, leggere il contenuto delle variabili che interessano e verificare che corrispondano alle aspettative.

La presente routine (lunga 153 byte), ogni volta che è "chiamata", visualizza il contenuto di tutte le variabili dichiarate fino a quel momento (escluse le matrici) senza arrestare il programma.

```
1000 PRINTCHR$(147)"QUESTA ROUTI
     NE PERMETTE DI VISUALIZZARE
     ,"
1010 PRINT"OGNI VOLTA CHE E' CHI
     AMATA,IL CONTENUTO"
1020 PRINT"DI TUTTE LE VARIABILI
      IN USO DEFINITE    FINO A Q
     UEL MOMENTO"
1030 PRINT"ESCLUSE LE MATRICI"
1040 RETURN
1050 DATA 165,122,133,253,165,12
     3
1060 DATA 133,254,165,45,133,251
     ,165,46,133,252
1070 DATA 169,255,133,122,169,1,
     133,123,160,0,177,251,201,1
     28,144, 32 ,41,127
1080 DATA 32,210,255,153,0,2,200
     ,177,251,41,127,240,7,32,21
     0,255,153,0,2
1090 DATA 200,169,37,32,210,255,
     153,0,2,208,42
1100 DATA 32,210,255,153,0,2,200
     ,177,251,240,32,201,128,144
     ,21
1110 DATA 41,127,240,7,32,210,25
     5,153,0,2,200,169,36,32,210
     ,255
1120 DATA 153,0,2,208,6,32,210,2
     55,153,0,2
1130 DATA 200,169,0,153,0,2,169,
     50,32,248,167,165,251,24,10
     5,7,133,251
1140 DATA 144,2,230,252,165,252,
     197,48,144,138,240,2,176,6,
     165,251,197,47
1150 DATA 144,128,165,253,133,12
     2,165,254,133,123,96,-1,200
     93
```

```
  0 REM DEMO
  1 REM HELP VARIABILI
  2 REM INIZIO LM=22782
  3 REM BY ARMANDO SFORZI
  4 :
100 DEF FNF(X)=RND(1)*10
110 FOR J=1 TO 5
120 A=FNF(0):X1=FNF(0):G%=FNF(0
    ):T$=STR$(FNF(0))
130 SYS22782:NEXT:END
```

Avvertenze

Ricordiamo ai nuovi lettori che ci leggono per la prima volta che la nostra rivista propone su ogni numero un gruppo di routine in Linguaggio Macchina per il Commodore 64, interamente rilocabili. Ciò significa che l'utente potrà realizzare una vera e propria enciclopedia in L.M. personalizzata in base alle proprie esigenze.

E' ovvio che le routine proposte, per funzionare adeguatamente, soggiacciono ad alcuni limiti ben illustrati nei primi numeri di "Nuovo Sistema".

In questa sede ci limitiamo a ricordare che, per evitare malfunzionamenti, è necessario:

- **Fissare il Top di memoria Ram a 20000.**
- **Caricare, servendosi del programma "Caricatore" a suo tempo pubblicato (o analoghi), le routine Basic contenenti le istruzioni Data.**
- **Indicare la prima locazione di memoria in cui si desidera allocare la routine stessa.**
- **Attivare la routine secondo i suggerimenti indicati nei Demo a corredo.**

Nuovo Sistema

a cura di Alessandro de Simone

Mappa della memoria di

NUOVO SISTEMA

(Elenco delle routine pubblicate)

Il primo valore indica l'indirizzo di partenza (coincidente con la SYS da impartire), mentre, il secondo, l'ultima locazione contenente l'ultimo dato.
Il numero fra parentesi, invece, si riferisce al numero di C.C.C. in cui sono state pubblicate le routine stesse.

20000/20011 GoTo Calcolato (31)
20012/20049 GoSub Calcolato (31)
20050/20128 Interp A$ (31)
20129/20188 Cambia colore (31)
20189/20245 Scroll Carattere (31)
20246/20302 Cancella caratt. (31)
20303/20445 GoSub Label (32)
20446/20562 GoTo Label (32)
20563/20596 Restore linea (33)
20597/20682 Disk Tool (33)
20683/20775 Directory (33)
20776/20858 Scroll Flag (34)
20859/20914 Deek (34)
20915/20952 Doke (34)
20953/21106 Decim/Esadec (35)
21107/21156 Locate cursor (35)
21157/21260 Beep (35)
21261/21473 Def.Sprite (36)
21474/21839 Sprite tool (36)
21840/21919 Colisione sprite (36)
21929/21962 Pause (36)
21963/22035 Cancella schermo (37)
22036/22174 Effetti sonori (37)
22175/22195 Up Scroll (37)
22196/22357 Right Scroll (37)
22358/22506 Left Scroll (37)
22507/22598 Down Scroll (37)

(Le routine di questo numero sono opera di Armando Sforzi)

Tre routine per chi inizia

A cura di Alessandro de Simone

16300 Print a velocità controllata
(Commodore 64/128)

Mediante questo sottoprogramma è possibile stampare qualsiasi messagio alfanumerico sullo schermo, ma in un modo diverso dal solito. Infatti le informazioni non vengono visualizzate di colpo sul video, ma gradualmente, carattere per carattere, grazie alla possibilità di impostazione della velocità di stampa.

Perfino la punteggiatura viene rispettata: infatti il programma introduce delle pause di ampiezza opportuna ogni qual volta incontra una virgola; inoltre, durante la stampa, il colore del bordo del video viene continuamente variato e, come se non bastasse, il tutto è accompagnato da un beep sonoro.

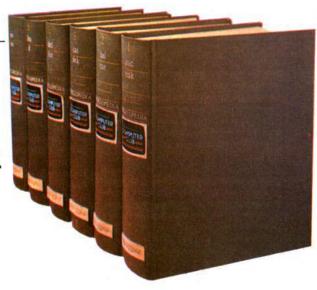
Qui di seguito elenchiamo le variabili da definire prima del Gosub, nonchè la loro funzione:

X1$: non contiene altro che i caratteri che volete stampare e, come tutte le stringhe, non deve possederne più di 255. Le informazioni verranno stampate nel colore che il cursore aveva al momento della "chiamata"; in X1$ possono essere inseriti anche i caratteri speciali (come se si trattasse di una normale Print);

X9: rappresenta la velocità di stampa. A valori bassi di X9 corrisponde velocità elevata (e viceversa). Se, al momento della chiamata, X9 è nulla verrà settata automaticamente la velocità ideale (cioè 10).

X1,Y1: contengono, rispettivamente, l'ascissa (X, variabile tra 0 e 39) e l'ordinata (Y: 0-24), ovvero le coordinate video alle quali si desidera venga stampato il primo carattere del messaggio.

La variabile X0$ ha funzione di controllo ed ha il compito di segnalare eventuali errori, come, ad esempio, i valori delle coordinate al di fuori del range stabilito (X0$="ERR").

```
100 REM  ESEMPIO D'USO
110 REM  PRINT A VELOCITA' CONT
    ROLLATA
120 REM  BY M. DELL'ABATE - SOL
    O C-64
130 :
140 PRINTCHR$(147)
150 X1$="PER MEZZO DI QUESTA RO
    UTINE HAI LA POSSIBILITA' D
    I STAMPARE "
160 X1$=X1$+"INFORMAZIONI SULLO
     SCHERMO, MA CON QUALCHE NO
    VITA':  "
170 X1$=X1$+"FACILITA' DI LETTU
    RA, PRESENTAZIONE ELEGANTE,
     VELOCITA' CONTRO"
180 X1$=X1$+"LLABILE E, COME PU
    OI VEDERE, ANCHE LA PUNTEGG
    IATURA VIENE RISP"
190 X1$=X1$+"ETTATA!"
200 X1=1:Y1=1:GOSUB 16300
210 END
9999 :
16300 X0$="OK":IF X1<0 OR X1>39 O
      R Y1<0 OR Y1>24 THEN X0$="E
      RR":RETURN
16320 IF X9<=0 THEN X9=10
16330 X2=LEN(X1$)
16340 POKE 211,Y1:POKE 214,X1:SYS
      58640
16350 FOR Y9=1 TO X2:Y1$=MID$(X1$
      ,Y9,1)
16360 POKE 53280,256*RND(1)
16370 PRINTY1$;:Y5=.2
16380 IF Y1$=CHR$(32) THEN Y5=4
16381 IF Y1$="," THEN Y5=7
16382 IF Y1$=";" OR Y1$=":" THEN
      Y5=10
16383 IF Y1$="." OR Y1$="!" OR Y1
      $="?" THEN Y5=15
```

```
16384 FOR Y3=0 TO X9*Y5:NEXT
16385 POKE 54296,15:POKE 54296,0:
      NEXT
16390 REM X1-Y1 COORDINATE
16391 REM X9 VELOCITA'
16392 REM X1$ STRINGA DA STAMPARE
16399 REM PRINT A VELOCITA' CONTR
      OLLATA
```

16400 Draw in bassa risoluzione
(Commodore 64/128)

Questa routine serve a tracciare semplici disegni sul video, segnalando gli spostamenti che una immaginaria "penna" compie sullo schermo. Si può scegliere il carattere da utilizzare ed il suo colore.

La variabile X1$ è la "stringa di definizione" cioè contiene i dati per il disegno; i simboli da utilizzare vengono qui di seguito elencati:

1 = alto
2 = basso
3 = sinistra
4 = destra
, = disegna
. = sposta la "penna" senza disegnare

Il punto (.) oppure la virgola (,) posti in coda alla stringa, indicano la fine del disegno.

Ad esempio, per disegnare un quadrato sarà sufficiente la sequenza:

X1$="2222444411113333."

mentre per disegnare una linea tratteggiata basterà assegnare:

X1$="44.44,44.44,44.44,44."

A X1 ed Y1 devono essere assegnate, prima del Gosub, le coordinate di inizio; in Y2 il codice (Poke di schermo) del carattere da utilizzare; in Y3 il suo colore (secondo la solita tabella 0-15 che si trova sul manuale di istruzione del computer).

```
100 REM  ESEMPIO D'USO
110 REM  DRAW IN BASSA RISOLUZI
    ONE
120 REM SOLO C-64. BY MAURIZIO
    DELL'ABATE
130 :
140 X1=1:Y1=1:Y2=81:Y3=1:PRINTC
    HR$(147)
150 X1$="22222444441144222224444
    2222222.44444411111111,44
    44444444444."
160 GOSUB 16400
170 FOR A=0 TO 1000:NEXT:PRINTC
    HR$(147)
180 PRINT" ORA PROVA TU....":PR
    INT
190 INPUT "POKECODE CARATTERE";
    Y2
200 INPUT "COLORE CARATTERE";Y3
210 INPUT "ASCISSA,ORDINATA";X1
    ,Y1
220 INPUT "STRINGA DI DEFINIZIO
    NE";X1$
230 PRINTCHR$(147):GOSUB 16400
240 PRINTCHR$(19);X0$:FOR A=0 T
    O 5000:NEXT: RUN
250 END
9999 :
15499 REM   DRAW IN BASSA RISOLUZ
      IONE
16400 X0$="OK":IF X1<0 OR Y1<0 OR
       X1>39 OR Y1>24 THEN X0$="E
      RR":RETURN
16410 IF Y2<0 OR Y2>255 OR Y3<0 O
      R Y3>255 OR X1$="" THEN X0$
      ="ERR":RETURN
16420 X2=LEN(X1$):Y4=1
16421 FOR Y7=1 TO X2
16422 X8=1024+X1+Y1*40
16423 IF X8<1024 OR X8>2023 OR Y4
      =0 THEN 16425
16424 POKE X8,Y2:POKE 54272+X8,Y3
16425 X2$=MID$(X1$,Y7,1)
16427 IF X2$="1" THEN Y1=Y1-1:GOT
      O 16450
16428 IF X2$="2" THEN Y1=Y1+1:GOT
      O 16450
16429 IF X2$="3" THEN X1=X1-1:GOT
      O 16450
16430 IF X2$="4" THEN X1=X1+1:GOT
      O 16450
16431 IF X2$="," THEN Y4=1:GOTO 1
      6450
16432 IF X2$="." THEN Y4=0:GOTO 1
      6450
16433 X0$="ERR":RETURN
16450 NEXT:RETURN
16490 REM X1$ STRINGA DI DEFINIZI
      ONE
16491 REM X1,Y1 ASCISSA,ORDINATA
      START
```

```
16492 REM Y2 CODICE CARATTERE DIS
      EGNO
16493 REM Y3 COLORE CARATTERE DIS
      EGNO
16499 REM DRAW IN BASSA RISOLUZIO
      NE
```

16500 Drum
(Commodore 64/128)

Drum, in inglese, vuol dire tamburo e questa routine ne simula il rumore e il ritmo. La sequenza di "suoni" viene definita in una variabile stringa in cui vengono utilizzati i seguenti caratteri-simbolo:

punto (.) = battito (bum)
spazio = pausa
meno (-) = doppia pausa

Tali informazioni vanno immesse in X1$ come si può notare nel demo in testa alla routine pubblicata.

Con un po' di pazienza riuscirete a comporre ritmi accettabili e, comunque, divertenti. La variabile numerica X2 deve contenere la velocità di esecuzione, tenendo conto che a valore basso corrisponde velocità elevata (30 è un valore valido nella maggior parte dei casi).

Occorre ancora precisare che in questa routine non si sono sfruttate le capacità sonore del SID: per generare il rumore è stato sufficiente alzare al massimo il volume e, subito dopo, riabbassarlo (registro Sid 54296).

```
100 REM ESEMPIO D'USO
110 REM DRUM. BY MAURIZIO DELL'
    ABATE
120 REM SOLO C-64
130 :
135 PRINTCHR$(147)
140 X1$=".- .- . . .- . . . .
    . .":X2=30:PRINTX1$:PRINT"V
    ELOC."X2:GOSUB 16500
150 FOR I=0 TO 900:NEXT
160 X1$=".- -.- -.... . . . .":
    X2=40:PRINTX1$:PRINTX2:GOSU
    B 16500
170 FOR I=0 TO 900:NEXT
180 X1$="... ... ...-. -... ...
     ....-..":X2=50:PRINTX1$:PR
    INTX2:GOSUB 16500
190 FOR I=0 TO 900:NEXT
200 X1$="............--. ..... .
    ... ....":X2=1:PRINTX1$:PRI
    NTX2:GOSUB 16500
210 END
```

```
 9999 :
16500 X0$="OK":X1=LEN(X1$):IF X2<
      0 THEN X0$="ERR":RETURN
16510 FOR Y9=1 TO X1
16515 XY$=MID$(X1$,Y9,1)
16520 IF XY$="." THEN GOSUB 16550
      :NEXT:GOTO 16540
16530 IF XY$="-" THEN FOR Y4=0 TO
       X2*10:NEXT:NEXT:GOTO 16540
16535 IF XY$=CHR$(32) THEN FOR Y4
      =0 TO X2*5:NEXT:NEXT:GOTO 1
      6540
16537 X0$="ERR"
16540 RETURN
16550 POKE 54296,15:POKE 54296,0:
      POKE 54296,15:POKE 54296,0
16560 FOR Y4=0 TO X2:NEXT:RETURN
16590 REM X1$ MELODIA
16591 REM X2  TEMPO
16599 REM DRUM
```

Elenco delle routine pubblicate

(Fra parentesi è riportato il numero di Commodore Computer Club su cui sono apparse)

16200 plot low-res (37)
16100 integrali (37)
16000 equaz. mista (37)
15900 equaz. terzo gr. (37)
15800 derivata di funz. (37)
15700 cornice rotante (37)
15600 convers. coordin. (36)
15500 logar. base quals. (36)
15400 conversione basi (36)
15300 semplif. frazioni (36)
15200 divis. con N decim.(36)
50100 directory (35)
15100 lampeggio righe (35)
15000 frammenta schermo (35)
14900 delete window (35)
14800 cambia stringhe (34)
14700 slitta stringhe (34)
14600 ruota stringhe (34)
10500 input programmab. (34)
14500 scroll solo testo (33)
14400 sprite multiuso (33)
14300 zoom esadecimale (33)
14200 video orologio (33)
11100 funzioni inverse (32)
13200 centra messaggi (32)
14100 finestre di testo (32)
14000 gestione nome disk (32)
13900 load/save pg.video (31)
13800 scritte in ebcm (31)
13700 bit image mps/803 (31)
13600 or esclusivo (31)
13500 comandi extra prg (31)
13400 linee low-res. (31)
13300 elabora stringhe (31)
13200 centratura frase (32)
13100 menu con joy (30)
13000 menu con cursore (30)
12900 frase lampeggiante (29)
12800 bordo technicolor (29)
12700 fill memoria ram (29)
12600 text copy mps 803 (29)
12500 colore pag.testo (29)
12400 print using (31)
12400 print using (29)
12300 m.c.d. E M.C.M. (29)
50500 visualizza file (28)
50400 read file relativi (28)
50300 write file relat. (28)
50200 crea file relativi (28)
50000 legge blocks free (28)
12200 numeri congrui (28)
12100 protezione S/W (28)
12000 koala (27)
11900 change pag. video (27)
11800 salva ram (27)
11700 calcolatrice (27)
11600 scomposiz.sillabe (27)
11500 car.hi-res (27)
11400 istogrammi (27)
50100 esame directory (26)
11300 funz.inv.iperbol. (26)
11200 funz.inv.trigonom. (26)
11100 funzioni inverse (26)
11000 funzioni iperbol. (26)
10900 convers. dec-esa (26)
10800 controllo data (25)
10700 impulsi sonori (25)
10600 reverse schermo (25)

Come realizzare l'enciclopedia e utilizzarla nei propri listati.

Ai lettori che hanno acquistato per la prima volta questo numero di Commodore Computer Club, illustriamo qui di seguito, in breve, i vantaggi derivanti dalla raccolta proposta. Questa, a pensarci bene, è la versione "superiore" della rubrica "1 RIGA" e potrebbe anche denominarsi... "Una schermata"!

Oltre che utili per costituire un'enciclopedia, i brevissimi sottoprogrammi pubblicati su ogni numero, sono anche validissimi strumenti di studio per coloro che desiderano approfondire le proprie conoscenze del Basic, esaminando, senza fatica, particolari routine o insolite tecniche di programmazione.

- Dato che può esser "chiamata" più di una volta nel corso di un programma, nessuna routine contiene istruzioni del tipo DATA oppure DIM, allo scopo di non creare confusione col listato principale.
- Nessuna routine può far riferimento ad altre routine dell'enciclopedia.
- Nessuna routine può contenere variabili "banali" (A, A$, eccetera), ma solo variabili poco usate (X1$, X8, Y0%, eccetera).
- Ogni routine deve apparire, **per intero,** sullo schermo del computer e consentire, proprio per questo motivo, di essere esaminata comodamente.
- Ogni routine deve esser numerata secondo uno standard che ha la particolarità di esser ricordato facilmente:

Righe	**Contenuto**
XXY00	Prima riga del sottoprogramma
XXY89	Ultima riga utile del sottoprogramma
XXY90 REM	Prima riga di spiegazioni
XXY99 REM	Nome della subroutine

in cui XX sono due valori variabili da 10 a 63; Y è un carattere numerico compreso tra 0 e 9.

Qualsiasi subroutine, in altre parole, inizia con un numero, di cinque caratteri, che termina **sempre** con "00". La stessa subroutine, d'altra parte, ha l'ultima riga numerata con "99". Digitando, ad esempio: LIST 10800-10899
si avrà la certezza di veder apparire sullo schermo, **per intero,** la routine il cui nome si trova nella riga 10899.

Prima di accedere alla routine, è necessario assegnare, alle variabili indicate con REM da riga XXY89 a XXY98, particolari valori per il suo corretto funzionamento. Al "ritorno" una o più variabili conterranno il risultato dell'elaborazione.

In questo modo, per esser più chiari, è possibile simulare alcuni comandi di versioni Basic avanzate oppure, addirittura, creare nuove e inedite istruzioni. Ad esempio, il comando: SOUND 1,800,500
che, nel C-16, riproduce un suono di tonalità 800 tramite la voce 1 per la durata 500, potrebbe venir riprodotta, in un'ipotetica subroutine per il Commodore 64, con: X1=1:X2=800:X3=500:GOSUB12400
nell'ipotesi, ovviamente, che la routine in oggetto sia allocata da riga 12400 a 12499.

I listati pubblicati "girano" su ogni computer, salvo dove indicato diversamente.

E' ovvio che nel caso del Vic-20, (che, come è noto, ha uno schermo di soli 506 caratteri), le subroutine "universali" funzionano correttamente, ma non possono apparire per intero in una sola schermata.

Per quanto riguarda la digitazione, si tenga presente che sulla rivista, per motivi di chiarezza, i comandi e le istruzioni Basic sono separati tra loro da spazi bianchi. Nel digitare le linee di programma, pertanto, è opportuno ignorarli altrimenti si rischia di non restare in una sola schermata. Se, per esempio, leggete:

```
12100 X1=34:  X2  = SQR(X3) + LOG(X1)
```

digitate nel modo seguente:

```
12100 X1=34:X2=SQR(X3)+LOG(X1)
```

senza, cioè, alcun carattere di separazione tra comandi ed istruzioni.

Collaborazione dei lettori

La collaborazione dei lettori è gradita, purchè si provveda a inviare **almeno** tre sottoprogrammi per volta, su nastro, disco oppure output di stampante. I listati di routine che non rispettano lo standard adottato non potranno esser presi in considerazione.

Tutti i lavori pubblicati verranno compensati con prodotti della Systems Editoriale (cassette di programmi, libri, abbonamenti, copie arretrate, eccetera).

Quella città in capo al mondo

Come determinare la distanza tra due città oppure come studiare, giocando, un capitolo di geografia o di geometria sferica

di Roberto Soro

Non staremo qui a richiamare le nozioni di geometria, indispensabili per lo studio approfondito di una parte fondamentale della geografia.

Ci limiteremo, invece, a darne semplici cenni in modo da facilitare la comprensione del programma.

Un po' di geografia

Come si può notare dal disegno, la Terra è idealmente suddivisa orizzontalmente in cerchi paralleli tra loro (detti, appunto, "paralleli"), e, verticalmente, in cerchi (Meridiani) che si incontrano tutti nei due Poli.

Esiste un solo parallelo più grande degli altri (equatore) che viene preso, quindi, come riferimento. Poichè, invece, i meridiani sono tutti eguali tra loro, per convenzione si assume come riferimento quello che passa per la città di Greenwich, in Inghilterra.

L'equatore, di latitudine zero, suddivide il pianeta in due semisfere: i paralleli posti al Nord sono considerati positivi, quelli al Sud negativi ed il valore assoluto massimo, ovviamente, è di 90 gradi.

Analogamente i meridiani posti ad Ovest di Greenwich (in senso, quindi, orario supponendo di appoggiare un orologio al Polo Nord) sono considerati positivi fino a 180 gradi; quelli posti ad Est, al contrario, sono considerati negativi, fino ad un massimo di 180 gradi.

Ne consegue che le città del nostro pianeta poste a Nord dell'equatore e ad Ovest di Greenwich hanno coordinate (dette più propriamente latitudine e longitudine) entrambe positive (America del Nord, gran parte dell'America centrale, parte dell'Africa Nord-Occidentale, Portogallo, parte della Spagna eccetera).

L'Italia, l'Europa Orientale, la Russia, il Giappone, l'India e gli altri Stati posti ad Est di Greenwich hanno una coordinata positiva ed una negativa, e così via.

Molto spesso, invece dei termini "positivo" e "negativo" si trova la simbologia O (Ovest) E (Est) per i meridiani e N (Nord) e S (Sud) per i paralleli.

Per inserire dati in un computer, e per elaborarli più facilmente, è molto più comodo inserire valori positivi e negativi anzichè valori preceduti (o seguiti) da N, S, E, O.

Come gira il programma

Anzitutto facciamo notare che, in coda al programma (da riga 1780 a 1990) sono inseriti i DATA relativi alle città. Il primo dato rappresenta il nome della città stessa, il secondo la sua latitudine ed il terzo la longitudine, positive o negative secondo quanto detto prima.

Il lettore potrà inserire tutte le città che desidera, a patto di rispettare la modularità appena accennata, rintracciando le informazioni utili in un qualsiasi atlante scolastico. Poichè il programma, dato il Run, determina automaticamente il numero di città inserite, non è necessario apportarvi alcuna variazione dal momento che l'ultimo Data (l'asterisco, riga 2000) comunica al computer il termine dei nomi.

Se, quindi, compare il messaggio "Out of data error" vorrà dire che avete dimenticato di digitare la riga 2000.

Dato il Run, dicevamo, compare un menu con quattro opzioni.

1/ Indovina le città
2/ Distanza per nomi
3/ Distanza per coordinate
4/ Elenco delle città

Le opzioni

Premuto il tasto "1" compaiono le iniziali di due città presenti in elenco, seguite, ciascuna, da un numero di puntini pari al numero dei caratteri nascosti. Compare, inoltre, la distanza in kilometri e la direzione in gradi. In base a tali informazioni l'utente deve indovinare, avendo a disposizione cinque tentativi per città, il loro nome.

Con la seconda opzione è possibile conoscere la distanza (in kilometri e gradi) esistente tra due qualsiasi città presenti in elenco. Nel caso una delle due sia assente, un opportuno messaggio comunica l'impossibilità di operare.

La terza opzione consente di conoscere la distanza tra due punti qualsiasi della Terra, fornendo le corrispondenti coordinate.

La quarta opzione si limita a visualizzare l'elenco delle città inserite nel programma.

FUSI ORARI

```
110 REM COMMODORE 64/128
120 REM C-16. PLUS 4
130 :
140 REM DISTANZE E DIREZIONI
150 REM AZIMUTALI SUL NOSTRO
160 REM PIANETA.
180 :
190 REM DI ROBERTO SORO (MILANO
    )
210 :
220 :
230 REM FUNZIONE ARCO COSENO
240 :
250 DEF FNCS(X)=-ATN(X/SQR(-X*X
    +1))+π/2
260 :
270 REM COSTANTI E VARIABILI
280 :
290 RO=π/180:REM CONVERSIONE GR
    ADI/RADIANTI
300 R=6367:REM RAGGIO MEDIO TER
    RESTRE
310 VD$=CHR$(17):VR$=CHR$(29):R
    EM TAB. VIDEO
320 VU$=CHR$(145):VH$=CHR$(19):
    REM TAB. VIDEO
330 V$=" " :VM$=VD$+VD$+VR$+VR$
    +VR$+VR$
340 VV$=VD$+VD$+VD$+VD$+VD$+VD$
    +VD$
350 :
360 REM INIZIALIZZAZIONE
370 :
380 CT=-1
390 READ A$:CT=CT+1:IF A$<>"*"G
    OTO 390
400 RESTORE
410 D=(CT/3)-1:DIM  TW$(D),CO(D
    ,1)
420 FOR I=0 TO D:READ  TW$(I),C
    O(I,0),CO(I,1):NEXT I
430 :
440 REM MENU PRINCIPALE
450 :
460 PRINTCHR$(147)SPC(12)"LE DI
    STANZE"
470 PRINTVM$+"1=INDOVINA LE CIT
    TA'"
480 PRINTVM$+"2=DISTANZA PER NO
    MI"
490 PRINTVM$+"3=DISTANZA PER CO
    ORDINATE"
500 PRINTVM$+"4=ELENCO DELLE CI
    TTA'"
510 PRINTVD$+VD$+VM$+"QUALE SCE
    LTA?";
520 GET J$:IF J$="" THEN  520
530 IF J$="1" THEN GOSUB 610:GO
    TO 460
540 IF J$="2" THEN GOSUB 830:GO
    TO 460
550 IF J$="3" THEN GOSUB 1050:G
    OTO 460
560 IF J$="4" THEN GOSUB 1130:G
    OTO 460
570 GOTO 520
580 :
590 REM 1 = SUBROUTINE  INDOVIN
    A
600 :
610 PRINTCHR$(147)SPC(8)"INDOVI
    NA LE DUE CITTA'"
620 PRINT
630 :
640 REM RICERCA DELLE DUE CITTA
    '
650 REM MEDIANTE NUMERI CASUALI
660 :
```

COORDINATE SFERICHE

Polo Nord
Longit. 0°
Longitudine
Parallelo di 50° Nord
Meridiano di 45° Est
Latitudine
Centro d. Terra
MERIDIANO FONDAMENTALE
EQUATORE = Latitudine 0
Asse Terrestre
Polo Sud


```
670 CA=INT(RND(0)*(D+1)):CB=INT
    (RND(0)*(D+1))
680 IF CA=CBGOTO 670
690 L1=CO(CA,0):M1=CO(CA,1)
700 L2=CO(CB,0):M2=CO(CB,1)
710 FOR I=1 TO LEN(TW$(CA))
720 PU$=".":IF I=1 THEN PU$=LEF
    T$(TW$(CA),1)
730 PRINTPU$;:NEXTI
740 PRINT:PRINTCHR$(145)SPC(20)
    ;
750 FOR I=1 TO LEN(TW$(CB))
760 PU$=".":IF I=1 THEN PU$=LEF
    T$(TW$(CB),1)
770 PRINTPU$;:NEXT I
780 GOSUB 1470:GOSUB  1620
790 PRINT:PRINT:PRINT:GOSUB 125
    0:GOSUB 1190:RETURN
800 :
810 REM 2 =  DISTANZE PER NOME
820 :
830 PRINTCHR$(147)SPC(9)"DISTAN
    ZE PER NOMI"
840 X$="":Y$="":CA=-1:CB=-1:PRI
    NTVH$+VD$+VD$
850 INPUT "QUALI CITTA' ( NOME1
     , NOME2 ) ";X$,Y$
860 IF (X$="") AND (Y$="")GOTO
    840
870 FOR I=0 TO D:REM LOOP DI CO
    NTROLLO NOMI
880 IF X$=TW$(I) THEN CA=I
890 IF Y$=TW$(I) THEN CB=I
900 NEXTI
910 IF (CA<>-1 AND CB<>-1) THEN
     950
920 IF CA=-1 THEN PRINTVD$+X$"N
    ON E' IN ELENCO"
930 IF CB=-1 THEN PRINTVD$+Y$"
    NON E' IN ELENCO"
940 GOTO 1010
950 PRINTCHR$(147)SPC(9)"DISTAN
    ZE PER NOMI"
960 PRINTVH$+VD$+VD$+X$
970 PRINTVU$;SPC(20):Y$
980 L1=CO(CA,0):M1=CO(CA,1)
990 L2=CO(CB,0):M2=CO(CB,1)
1000 GOSUB 1470:GOSUB 1620
1010 GOSUB 1190:RETURN
1020 :
1030 REM 3 = DISTANZE PER COORDI
     NATE
1040 :
1050 PRINTCHR$(147)SPC(6)"DISTAN
     ZE PER COORDINATE"
1060 CL$="COORDINATE: LAT , LONG
     : "
1070 PRINTVH$+VD$+VD$+CL$;:INPUT
      L1,M1
1080 PRINTCL$;:INPUT L2,M2
1090 GOSUB 1470:GOSUB 1620:GOSUB
      1190:RETURN
1100 :
1110 REM 4 = ELENCO  CITTA'
1120 :
1130 PRINTCHR$(147)SPC(7)"ELENCO
      DELLE" D+1 "CITTA'"VD$
1140 FOR I=0 TO D:PRINTTW$(I),;:
     NEXT I
1150 GOSUB 1190:RETURN
1160 :
1170 REM SUBROUTINE DI ATTESA
1180 :
1190 PRINT:PRINTVD$+VD$+"PREMI U
     N TASTO"
1200 GET J$:IF J$="" THEN 1200
1210 RETURN
1220 :
1230 REM CONTROLLO NOMI CITTA'
1240 :
1250 NC=0:NT=0:C1=0:C2=0
1260 PRINTCHR$(145);
1270 FOR K=1 TO 39:PRINTV$;:NEXT
     K:PRINT
1280 PRINTVU$"NOME DI UNA CITTA'
     ";
1290 INPUT X$:NT=NT+1
1300 IF X$=TW$(CA) THEN SW=1:NC=
     NC+1:C1=1:GOSUB 1400
1310 IF X$=TW$(CB) THEN SW=2:NC=
     NC+1:C2=1:GOSUB 1400
1320 IF NC=2 THEN RETURN
1330 IF NT=5 AND NC=0 THEN SW=1:
     NC=1:C1=1:GOSUB 1400
1340 X$=" ":IF NT<10 THEN GOTO 1
```

MERIDIANI FONDAMENTALI

```
     260
1350 SW=1:IF SW=C1 THEN SW=2
1360 GOSUB 1400:X$=" ":RETURN
1370 :
1380 REM STAMPA NOMI INDOVINATI
1390 :
1400 IF SW=1 THEN X$=TW$(CA):CH=
     0
1410 IF SW=2 THEN X$=TW$(CB):CH=
     20
1420 PRINTVH$+VD$+VD$SPC(CH)X$:P
     RINTVV$
1430 RETURN
1440 :
1450 REM CALCOLO DELLA DISTANZA
1460 :
1470 LX=M1-M2:K1=90-L1:K2=90-L2
1480 XT=SIN(K1*R0)*COS(M1*R0)
1490 XR=SIN(K2*R0)*COS(M2*R0)
1500 YT=SIN(K1*R0)*SIN(M1*R0)
1510 YR=SIN(K2*R0)*SIN(M2*R0)
1520 ZT=COS(K1*R0):ZR=COS(K2*R0)
1530 G=XT*XR+YT*YR+ZT*ZR
1540 IF G>=1 THEN G=1:GA=0:GOTO
     1570
1550 IF G<=-1 THEN G=-1:GA=π:GOT
     O 1570
1560 GA=FNCS(G)
1570 DK=INT(R*GA)
1580 PRINT:PRINTVD$+VD$"DISTANZA
        "DK" KM":RETURN
1590 :
1600 REM CALCOLO DELLA DIREZIONE
1610 :
1620 A3=SIN(GA)*SIN(K1*R0)
1630 IF A3=0 THEN A3=1E-20
1640 C=(ZR-G*ZT)/A3
1650 IF C>=1 THEN C=1:AZ=0:GOTO
     1680
1660 IF C<=-1 THEN C=-1:AZ=π:GOT
     O 1680
1670 AZ=FNCS(C)
1680 AZ=AZ/R0:REM CONVERSIONE IN
      GRADI
1690 T1=LX<0 AND LX>=-180:T2=LX>
     180 AND LX<=360
1700 IF T1 OR T2 THEN AZ=360-AZ
1710 AZ=INT(AZ*10)/10
1720 PRINTVD$"DIREZIONE  "AZ" GR
     ADI":RETURN
1730 :
1740 :
1750 REM CITTA' E LORO COORDINAT
     E GEOGRAFICHE
1760 REM SE IL NOME DI UNA CITTA
     ' E' COMPOSTO DA DUE PAROLE
     , RICORDARSI
1761 REM DI RACCHIUDERLI TRA API
     CI. ES: "NEW YORK"
1770 :
1780 DATA "BUENOS AIRES",-34.7,5
     8.4
1790 DATA CAPETOWN,-34.0,-18.5
1800 DATA "HONG KONG",22.5,-114.
     2
1810 DATA LIMA,-12.0,77.2
1820 DATA LONDRA,51.5,0.0
1830 DATA "LOS ANGELES",34.0,118
     .3
1840 DATA MADRID,40.5,3.7
1850 DATA MELBOURNE,-37.8,-145.0
1860 DATA "MEXICO CITY",19.5,99.
     2
1870 DATA MILANO,45.5,-9.1
1880 DATA MOSCA,55.8,-37.7
1890 DATA NAIROBI,-1.5,-36.8
1900 DATA "NEW YORK",40.8,74.0
1910 DATA OUAGADOUGOU,12.2,2.6
1920 DATA PARIGI,48.9,-2.3
1930 DATA PECHINO,39.9,-116.5
1940 DATA PRAGA,50.1,-14.4
1950 DATA "RIO DE JANEIRO",-22.9
     ,43.3
1960 DATA ROMA,41.8,-12.5
1970 DATA "SAN FRANCISCO",37.7,1
     22.4
1980 DATA SYDNEY,-33.8,-151.2
1990 DATA TOKYO,35.7,-139.7
1991 :
2000 DATA *
2001 :
2010 REM ATTENZIONE ALL'ULTIMO D
     ATA (*)
2020 END
```

Risolutore di crucipuzzle

Un programma di notevole utilità per partecipare a concorsi che mettono in palio premi di tutto rispetto

di Pietro Pino

Ricorderemo brevemente che cosa è un crucipuzzle ai pochi lettori che forse ancora non lo sanno.

Come si può vedere dalla figura, un crucipuzzle è una sorta di cruciverba in cui sono presenti caratteri alfabetici sparsi senza apparente relazione tra loro.

Per risolvere il puzzle è necessario contornare il gruppo di caratteri che rappresentano i nomi riportati nello stesso indovinello in un qualsiasi modo, purchè rettilineo: orizzontale (da destra a sinistra o viceversa), verticale o diagonale. Alla fine delle operazioni rimangono escluse alcune lettere che, lette di seguito, rappresentano la cosiddetta "chiave" del crucipuzzle.

L'esempio riportato, corrispondente ad una griglia di 10x7 è piuttosto semplice da indovinare, ma esistono schemi ben più complessi.

Il programma pubblicato in queste pagine, autoesplicativo, è in grado di risolvere crucipuzzle fino ad una dimensione massima di 39x24 e consente, entro certi limiti, eventuali correzioni e manipolazioni.

Il lettore potrà inserire la possibilità di memorizzare su supporto magnetico lo schema introdotto (la cui digitazione richiede una fatica pro-

porzionale alla sua dimensione) in modo da richiamarlo a volontà in caso di errori o di interruzione, specialmente in fase di debug.

A che cosa può servire un programma che, in pratica, vi priva del divertimento peculiare dei cruciverba? Ma è ovvio: avere più chances nella partecipazione ai concorsi...

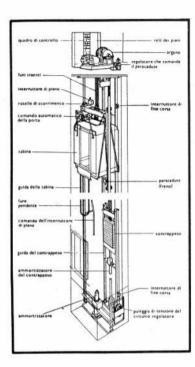
```
100 REM CRUCIPUZZLE
110 REM PER C/64
120 REM BY PIETRO PINO
130 REM MILANO
140 :
150 GOSUB 1720:GOSUB 1490:GOTO
    1220
160 :
170 REM INSERISCI DIAGRAMMA
180 :
190 PRINT"[CLEAR]";:X$="VUOI UN
     NUOVO DIAGRAMMA (S/N)":GOS
    UB 1440
200 GET A$:IF A$<>"S" AND A$<>"
    N" THEN 200
210 IF A$="S" THEN 240
220 X$="ALLORA SCEGLI L'OPZIONE
     2!":GOSUB 1440:FOR T=1 TO
    1000:NEXT:GOTO 1220
230 :
240 CLR :GOSUB 1490
250 INPUT "[CLEAR]LARGHEZZA DEL
    LA GRIGLIA";LA
260 IF LA<1 OR LA>39 THEN 250
270 INPUT "[7 DOWN]ALTEZZA DELL
    A GRIGLIA";AL
280 IF AL<1 OR AL>24 THEN 270
290 PRINT"[NERO][CLEAR]";
300 X$="QUANDO HAI FINITO PREMI
     'RETURN'":GOSUB 1440
310 FOR X=1 TO LA:A$=A$+".":NEX
    T
320 FOR Y=1 TO AL:PRINTA$:NEXT
330 DIM L$(AL-1,LA-1)
340 PRINT"[HOME]";:SYS(65487):X
    $="ATTENDI":GOSUB 1440
350 FOR Y=0 TO AL-1:FOR X=0 TO
    LA-1
360 P=PEEK(1024+Y*40+X):IF P<1
    OR P>26 THEN P=-18
370 P=P+64
380 L$(Y,X)=CHR$(P):NEXT:NEXT:G
    OTO 1220
390 :
400 REM INSERISCI LISTA
410 :
420 PRINT"[CLEAR]";:IF F2=0 THE
    N 440
430 X$="HAI GIA' INSERITO LA LI
    STA":GOSUB 1440:FOR T=1 TO
    1000:NEXT:GOTO 1220
440 INPUT "QUANTI NOMI DEVI INS
    ERIRE";A
450 IF A<1 THEN 440
460 DIM N$(A-1):F2=1:PRINT"[CLE
    AR][2 DOWN]"
470 FOR I=0 TO A-1
480 PRINT"NOME"I+1;
490 INPUT N$(I):IF N$(I)="" OR
    LEN(N$(I))<4 THEN 480
500 NEXT:GOTO 1220
510 :
520 REM INIZIA LA RICERCA
530 :
540 PRINT"[CLEAR]";:X$="":IF F2
    =0 THEN X$="PRIMA DEVI INSE
    RIRE LA LISTA"
550 IF L$(0,0)="" THEN X$="PRIM
    A DEVI INSERIRE IL DIAGRAMM
    A"
560 IF X$="" THEN 580
570 GOSUB 1440:FOR T=1 TO 1000:
    NEXT:GOTO 1220
580 GOSUB 1000
590 FOR K=0 TO A-1:B$=N$(K):IF
    PEEK(653)=1 THEN 620
600 X$="PREMI SHIFT":GOSUB 1440
610 IF PEEK(653)<>1 THEN 610
620 X$=B$:GOSUB 1440
630 I$=LEFT$(B$,1)
640 FOR Y=0 TO AL-1:FOR X=0 TO
    LA-1
650 IF I$=L$(Y,X) THEN 690
660 NEXTX:NEXTY
```

```
670 X$="IL NOME"+STR$(K+1)+" NO
    N ESISTE NEL DIAGRAMMA":GOS
    UB 1440:
680 FOR T=1 TO 800:NEXT:GOTO 98
    0
690 L=LEN(B$)-1
700 N=1:S=1:E=1:O=1:NE=1:SE=1:N
    O=1:SO=1
710 IF (X+L)>LA-1 THEN E=0:NE=0
    :SE=0
720 IF (X-L)<0 THEN O=0:NO=0:SO
    =0
730 IF (Y+L)>AL-1 THEN S=0:SE=0
    :SO=0
740 IF (Y-L)<0 THEN N=0:NE=0:NO
    =0
750 IF N=1 THEN VX=0:VY=-1:GOSU
    B 850:IF FL<>0 THEN 930
760 IF S=1 THEN VX=0:VY=1:GOSUB
     850:IF FL<>0 THEN 930
770 IF E=1 THEN VX=1:VY=0:GOSUB
     850:IF FL<>0 THEN 930
780 IF O=1 THEN VX=-1:VY=0:GOSU
    B 850:IF FL<>0 THEN 930
790 IF NE=1 THEN VX=1:VY=-1:GOS
    UB 850:IF FL<>0 THEN 930
800 IF NO=1 THEN VX=-1:VY=-1:GO
    SUB 850:IF FL<>0 THEN 930
810 IF SE=1 THEN VX=1:VY=1:GOSU
    B 850:IF FL<>0 THEN 930
820 IF SO=1 THEN VX=-1:VY=1:GOS
    UB 850:IF FL<>0 THEN 930
830 GOTO 660
840 :
850 REM RICERCA
860 :
870 X1=X:Y1=Y:FL=0
880 FOR V=2 TO L+1:X1=X1+VX:Y1=
    Y1+VY:M$=MID$(B$,V,1)
890 IF M$=L$(Y1,X1) THEN NEXT
900 IF V=L+2 THEN FL=1
910 RETURN
920 :
930 REM VISUALIZZA PAROLA TROVA
    TA
940 :
950 FOR I=0 TO L:P=1024+Y*40+X:
    IF P>2024 OR P<1024 THEN P=
    1024
960 POKE P,(PEEK(P) AND 63) OR
    64:POKE P+54272,0:REM L$(Y,
    X)="▨"+L$(Y,X)+"WAIT "
970 X=X+VX:Y=Y+VY:NEXT
980 NEXTK:GOTO 1070
990 :
1000 REM STAMPA DIAGRAMMA
1010 :
1020 PRINT"[CLEAR][NERO]";
1030 FOR Q=0 TO AL-1:FOR W=0 TO
     LA-1
1040 PRINTL$(Q,W);:NEXT:PRINT:NE
     XT
1050 RETURN
1060 :
1070 REM TROVA LA CHIAVE
1080 :
1090 C$="":X$="PREMI 'C' PER AVE
     RE LA CHIAVE":GOSUB 1440
1100 GET A$:IF A$<>"C" THEN 1100
1110 X$="ATTENDI":GOSUB 1440
1120 FOR Q=0 TO AL-1:FOR W=0 TO
     LA-1
1130 P=PEEK(1024+Q*40+W)
1140 IF P<>46 AND ((P AND 192)=0
     ) THEN C$=C$+CHR$(P+64)
1150 NEXT:NEXT:IF C$="" THEN C$=
     "?"
1160 PRINT"[HOME]";:FOR I=1 TO 2
     4:PRINTCL$:NEXT
1170 PRINT"[HOME][12 DOWN]CHIAVE
     : "C$
1180 X$="PREMI LA BARRA PER TORN
     ARE AL MENU":GOSUB 1440
1190 GET A$:IF A$<>" " THEN 1190
1200 GOTO 1220
1210 :
1220 REM MENU
1230 :
1240 POKE 53280,12:POKE 53281,12
     :POKE 646,14
1250 POKE 53265,91:POKE 53282,5:
     POKE 53283,2:POKE 53284,5
1260 PRINT"[NERO][CLEAR][2 DOWN]
     " TAB(9)"* * *  M E N U  *
     * *"
1270 PRINT"[3 DOWN]  [1].....INS
     ERISCI IL DIAGRAMMA"
1280 PRINT"[2 DOWN]  [2].....COR
     REGGI IL DIAGRAMMA"
1290 PRINT"[2 DOWN]  [3].....INS
```

```
     ERISCI LA LISTA DI PAROLE"
1300 PRINT"[2 DOWN]  [4].....COR
     REGGI LA LISTA DI PAROLE"
1310 PRINT"[2 DOWN]  [5].....RIS
     OLVE IL DIAGRAMMA"
1320 PRINT"[3 DOWN]" TAB(15)"- S
     CEGLI -"
1330 GET A$:IF A$="" THEN 1330
1340 IF A$<"1" OR A$>"5" THEN 13
     30
1350 ON VAL(A$)GOTO 170,1370,400
     ,1570,520
1360 :
1370 REM CORREGGI DIAGRAMMA
1380 :
1390 PRINT"[CLEAR]";:IF L$(0,0)<
     >"" THEN 1420
1400 X$="PRIMA DEVI INSERIRE IL
     DIAGRAMMA":GOSUB 1440:FOR T
     =1 TO 1000:NEXT
1410 GOTO 1220
1420 GOSUB 1000:X$="CORREGGI IL
     DIAGRAMMA":GOSUB 1440:GOTO
     340:
1430 :
1440 REM STAMPA MESSAGGIO IN BAS
     SO
1450 :
1460 PRINT"[HOME]":PRINTV$CL$H$X
     $RIGHT$(S$,42-LEN(X$))H$"[H
     OME][RVOFF]";
1470 RETURN
1480 :
1490 REM DEFINISCE LE STRINGHE
1500 :
1510 V$="[NERO][HOME][24 DOWN][R
     VS]":REM 1 HOME + 24 CRSR-D
     OWN + 1 RVS-ON
1520 S$="
                         [LEFT][INST
     ] "
1530 H$="[39 LEFT]"
1540 CL$="[RVOFF][RIGHT]"+CHR$(2
     0)+"
                         [RVS]"
1550 RETURN
1560 :
1570 REM CORREGGE LISTA
1580 :
1590 PRINT"[CLEAR]";:IF F2<>0 TH
     EN 1620
1600 X$="PRIMA DEVI INSERIRE LA
     LISTA":GOSUB 1440
1610 FOR T=1 TO 1000:NEXT:GOTO 1
     220
1620 PRINT"LA LISTA IN MEMORIA E
     ' FORMATA DA"A:PRINT"PAROLE
     [DOWN]"
1630 PRINT"SE LA PAROLA VA BENE,
      PREMI 'S'.          VICEVERS
     A, PREMI 'N'.[DOWN]"
1640 FOR I=0 TO A-1
1650 PRINT"NOME"I+1": "N$(I);
1660 GET A$:IF A$<>"S" AND A$<>"
     N" THEN 1660
1670 IF A$="S" THEN PRINT:GOTO 1
     690
1680 INPUT N$(I)
1690 NEXTI
1700 GOTO 1220
1710 :
1720 REM SPIEGAZIONI
1730 PRINT"[CLEAR][2 DOWN]   * *
     *  C R U C I P U Z Z L E   *
      * *"
1740 PRINT"[2 DOWN] I CRUCIPUZZL
     E SONO QUEI GIOCHI "
1750 PRINT" ENIGMISTICI CONSISTE
     NTI NEL RICERCARE "
1760 PRINT" DELLE PAROLE INSERIT
     E ORIZZONTALMENTE"
1770 PRINT" VERTICALMENTE O DIAG
     ONALMENTE "
1780 PRINT" ALL'INTERNO DI UN DI
     AGRAMMA."
1790 PRINT"[2 DOWN] DOPO AVER CA
     NCELLATO TUTTE LE PAROLE, "
1800 PRINT" ALLA FINE RIMARRANNO
      DELLE LETTERE "
1810 PRINT" CHE LETTE DI SEGUITO
      DARANNO LA CHIAVE."
1820 PRINT"[2 DOWN] QUESTO PROGR
     AMMA RISOLVE QUALSIASI "
1830 PRINT" CRUCIPUZZLE FORNENDO
     NE LA CHIAVE."
1840 PRINT"[2 DOWN]" TAB(13)"[RV
     S]PREMI UN TASTO[RVOFF]"
1850 WAIT 197,191:POKE 198,0
1860 RETURN
1870 END
```

L'ascensore

Una nuova sfida per i nostri lettori: avete senso logico?

a cura di Alessandro de Simone

La precedente "sfida" ha raccolto numerose adesioni e ci ha spronato a lanciarne altre. I Pangrammi (CCC N.33), purtroppo, sono stati forse considerati un po' (troppo) impegnativi e le proposte pervenute in Redazione non erano di livello soddisfacente.

Eccoci dunque a parlare di qualcosa di più semplice ma, non per questo, meno valido da un punto di vista strettamente logico-informatico.

La logica di un ascensore

Avete mai riflettuto sulla struttura elettromeccanica che sovrintende al funzionamento di un ascensore? Quando, nell'uso quotidiano, premiamo uno dei suoi pulsanti, tutto ci appare molto semplice ma, in realtà, è necessario che una serie di operazioni logiche, strettamente correlate tra di loro, gestiscano movimenti e meccanismi di notevole complessità.

Ecco dunque la proposta di scrivere, in Basic, un programma che, scandendo la tastiera, faccia muovere un ascensore sullo schermo e lo arresti o lo faccia proseguire nella corsa a seconda delle richieste effettuate da ipotetici utenti.

Che cosa deve fare il programma

Supponendo di installare un ascensore in una palazzina di (almeno) cinque piani, dovrete assegnare, per ciascun piano, due pulsanti (=tasti) di "chiamata": uno per la salita e l'altro per la discesa. Altri sette tasti simuleranno altrettanti pulsanti presenti in cabina: uno per ciascun piano oltre a quello di Alt e di Allarme. Ciò premesso dovrete, in definitiva, scrivere un programma in grado di:

- **far muovere l'ascensore dal punto in cui si trova al piano di chiamata.**
- **memorizzare, in ordine, le fermate ai vari piani in base alle richieste che si verificano anche durante lo spostamento tra un piano e l'altro.**
- **individuare le chiamate "doppie" ed annullarle in modo da evitare soste prolungate al piano**
- **attendere alcuni secondi prima di richiudere le porte e riaprirle nel caso in cui un utente ne impedisca la chiusura.**
- **simulare le operazioni di arresto in caso di Stop e di allarme.**
- **verificare l'eventuale sovraccarico ed impedirne la partenza.**
- **spegnere le luci interne nel caso di prolungata inattività.**

E' ovvio che saranno privilegiati gli utenti del Commodore 64 che, grazie agli Sprite, riusciranno più facilmente a gestire le animazioni. I possessori del C/128, poi, dovrebbero realizzare listati più che divertenti.

Saranno comunque privilegiati per la pubblicazione soprattutto programmi molto brevi e, magari, trasformati in giochi. L'ascensore e le sue porte, quindi, simulateli con rettangoli "pieni" di bit (decimale 255) in modo da disegnarli con un For... PokeX,255... Next.

I più apprezzati in assoluto, comunque, saranno i listati facilmente comprensibili, pieni di REM e "accessibili" (=modificabili) dai nostri lettori, anche se principianti.

Per standardizzare i programmi, che ci auguriamo giungano numerosi, attenetevi alle seguenti... imposizioni:

- **i pulsanti presenti in cabina simulateli con i tasti da 1 a 9; per l'allarme usate il tasto di freccia a sinistra mentre per l'Alt usate il tasto più (+).**
- **la fila dei tasti Q W E R... utilizzateli per simulare la chiamata per la discesa dei rispettivi piani 1, 2, eccetera, mentre per la prenotazione in salita riferitevi ai tasti A, S, D...**
- **per il movimento dei passeggeri servitevi di un joystick o dei tasti funzione.**

E' INDISPENSABILE inviare i listati su supporto magnetico (disco o nastro) in modo da consentirne l'esame con maggiore velocità.

L'indirizzo è il solito:

SYSTEMS Editoriale
Commodore Computer Club
(L'ascensore)
Viale Famagosta, 75
20142 Milano

Due programmi... ipnotici!

Semplicissime quanto valide applicazioni delle arcinote routine grafiche di Toma

di Paolo Agostini

Come l'attento lettore avrà notato, da un po' di tempo la Systems Editoriale si presenta in edicola con prodotti che rappresentano una risposta alle pressanti richieste dei lettori.

Le famose routine grafiche di Toma sono state riunite in un unico fascicolo di "Commodore Speciale" (tra breve in edicola), ampliate, aggiornate, commentate e precedute da un vero e proprio mini-corso di Linguaggio Macchina.

Per venire incontro ai lettori, vecchi e nuovi, appassionati di grafica, ecco pronti due listati semplicissimi in grado di tracciare un'infinità di curve modificando semplicemente i parametri in ingresso.

Gli intervalli dei valori da introdurre sono piuttosto ampi e modificabili a volontà, ma per ciascun Input ne sono suggeriti alcuni allo scopo di prender confidenza con i listati.

```
 90 REM  HI-RES PER C/64
 92 REM  E' INDISPENSABILE CARI
    CARE E LANCIARE PRIMA LE RO
    UTINE DI TOMA
 93 :
100 CLR :PRINTCHR$(147)
110 INPUT "N.PUNTI (5-30)";N
120 INPUT "RAGGIO (20-100)";R
125 INPUT "CANCELLO HI-RES (S/N
    )";CC$
127 INPUT "UNA LINEA PER VOLTA
    (S/N)";LL$
130 D=2*π/N:DIM X(N),Y(N)
140 FOR I=1 TO N:T=T+D
150 X(I)=R*COS(T):Y(I)=R*SIN(T)
    :NEXT
160 S=N-1
170 IF CC$="S" THEN :←CLEAR
175 ←COLOR 1: ←GRAF0,1
180 FOR I=1 TO S:Z=I+1
190 FOR J=Z TO N
195 IF LL$="S" THEN  GOSUB  240
200 ←DRAW X(I),Y(I),0,X(J),Y(J)
    ,0
210 NEXTJ:NEXTI
220 POKE 198,0:WAIT 198,1:POKE
    198,0
230 ←TEXT 1,12:GOTO 100
240 GET A$:IF A$="" THEN 240
250 RETURN
```

```
 90 REM  SPIRALI PER C/64 & ROU
    TINE DI TOMA
 91 REM  BY PAOLO AGOSTINI
 92 :
100 PRINTCHR$(147)
110 INPUT "NUMERO DI SPIRALI (1
    -5)";N
120 INPUT "RAGGIO/ANGOLO INIZ.(
    20-100)";R
130 NN=N*100:RA=R/NN:DIM T(NN):
    A=A1
140 ←CLEAR: ←COLOR 1: ←GRAF 0,
    1
150 FOR J=1 TO N:A=A+2*π/N
160 I=A:TH=2*π/100:T(0)=T
170 FOR I=1 TO NN:T(I)=T(I-1)+T
    H:RR=RA*I
180 X1=(RR-RA)*COS(T(I-1))
185 Y1=(RR-RA)*SIN(T(I-1))
190 X2=RR*COS(T(I)):Y2=RR*SIN(T
    (I))
195 ←DRAW X1,Y1,0,X2,Y2,0
200 NEXTI
210 NEXTJ
220 POKE 198,0:WAIT 198,1:POKE
    198,0
230 ←TEXT 1,12:POKE 53280,2
```

RICHIESTA ARGOMENTI

Mi farebbe piacere che Commodore Computer Club parlasse più spesso dei seguenti argomenti:

1/ ....

2/ ....

3/ ....

4/ ....

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI ....

....

....

....

....

....

CERCO/OFFRO CONSULENZA ....

....

....

....

....

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... Cognome ....

Via .... N° .... CAP. .... Città ....

Telefono .... Orario ....

Entra nel grande Club

Fin dallo sbarco in Italia della Commodore ***Commodore Computer Club*** è il punto di riferimento di tutti gli utenti di C/64, Vic 20, C/16, Plus 4 ed ora di PC 10/20 ed Amiga.

Articoli didattici, recensioni e programmi istruttivi ed a basso costo hanno fatto di *Commodore Computer Club* la prima rivista italiana d'informatica.

Ma, per i lettori, Commodore Computer Club non è solo rivista: è consulenza telefonica gratuita, software originale pubblicato a latere dalla stessa casa editrice, un ponte verso l'informatica "maggiore" anche attraverso la collaborazione con le riviste sorelle *"Personal Computer"* e *"Computer"*.

E' per questa ragione che, anno dopo anno, aumenta il numero dei lettori che preferiscono ricevere la rivista in abbonamento invece di acquistarla in edicola. Ad essi l'editore riserva una serie di vantaggi esclusivi come:

- **un libro in omaggio** da scegliere tra i titoli disponibili della collana I libri di Systems*;
- **l'uso di una linea telefonica speciale** per richieste di consigli, e consulenza, il cui numero e le modalità d'uso verranno comunicate in forma riservata alla ricezione dell'abbonamento;
- **un canone annuo particolarmente interessante** di lire 40.000 per 11 fascicoli di Commodore Computer Club e di lire 35.000 per 11 fascicoli di Personal Computer;
- **l'esclusivo canone cumulativo** di lire 65.000 per 11 fascicoli di Commodore Computer Club ed 11 di Personal Computer;
- **uno sconto del 10%** su tutti gli acquisti per corrispondenza dei prodotti software su disco o cassetta, fascicoli arretrati o libri della Systems senza limiti di quantità.

** I titoli disponibili sono quelli reclamizzati sull'apposita pagina pubblicitaria "La libreria di Systems".*

Inviatemi in omaggio il volume della collana i libri di Systems..

Registrate oggi stesso il mio abbonamento a: ☐ Commodore Computer Club (Lire 40.000)
☐ Commodore Computer Club + Personal Computer (Lire 65.000)

☐ Desiderando ricevere le copie ordinate con la massima urgenza, accludo assegno bancario n.ro....................
Banca.................................... per lire.................................... voi intestato.

☐ Contentandomi dei normali tempi postali ho inviato oggi stesso l'importo di lire .... a mezzo C/C postale N. 37952207 intestato a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

Nome ....
via .... **N.ro.** .... **telefono** ....
CAP .... **Città** ....

Ritagliare e spedire in busta chiusa regolarmente affrancata a Systems Editoriale - V.le Famagosta, 75 - 20142 Milano.

LEGGO VR
PERCHÈ
HO UN'IDEA
FISSA
IN TESTA
Il lettore di *VR Videoregistrare* è giovane, dinamico, creativo. Di cultura e reddito superiore alla media, possiede spesso più di un videoregistratore, oltre all'impianto hi-fi e al computer: nel tempo libero, non rinuncia a viaggi in Italia e all'estero, e a cinema, teatro e spettacoli sportivi in genere. Usa il videoregistratore non solo per i programmi tv o preincisi, ma anche per riprendere i momenti felici in famiglia, per creare una videoteca personale. E tu, che lettore sei?
WIZ 2 - IMMAGINE TERRY NIEDZIALEK - NYC
VR
VIDEOREGISTRARE
L'immaginazione al potere

La funzione Random: applicazioni pratiche

Una trattazione approfondita, corredata di numerosi programmi, di una delle classiche applicazioni informatiche: la casualità e la conseguente "Fortuna"

di Alessandro de Simone

Da centinaia di anni, in pratica da quando l'uomo ha iniziato a ragionare, molti eventi, piacevoli o meno, sono stati attribuiti alla Sorte, spesso trascurando la responsabilità di coloro che, in un modo o in un altro, hanno partecipato a provocarli.

A volte, infatti, si attribuisce alla "Sfortuna" un incidente automobilistico determinato da pura, scientifica incoscienza; altre volte si identifica nei gatti neri, o nelle scale a pioli, la causa di avversità capitate.

La mentalità, davvero comoda, di assegnare ad altri responsabilità proprie, ha provocato, e provoca tuttora, il fiorire di giochi basati sulla fortuna e sulla casualità, permettendo, come inevitabile conseguenza, il diffondersi di scommesse di ogni tipo.

La creduloneria della gente ha consentito, altresì, la possibilità di "manipolare" a proprio vantaggio alcuni giochi e l'elenco delle truffe possibili non è ancora completo.

Molto spesso, d'altra parte, un esame obiettivo dei fenomeni e delle loro cause è alla base di interessanti quanto utili ricerche scientifiche: conoscere il numero di ammalati nel caso si determinino certe condizioni, determinare la probabilità stessa del verificarsi di dette condizioni, non può che essere utile per la prevenzione nel campo medico.

Prevedere, analogamente, il numero di disoccupati nel caso si verifichi una nuova crisi del petrolio, e proporre l'ideale "riciclo" dei lavoratori, è uno dei problemi più studiati di moderni economista.

Ecco, dunque, l'importanza di esaminare da vicino il principale compito che un computer può svolgere: generare in successione numeri casuali, non affetti, cioè, da alcun legame tra loro.

Che cosa è il caso?

Quando si inizia lo studio del calcolo delle probabilità è quasi inevitabile riferirsi ad un oggetto che tutti noi abbiamo visto, e adoperato, almeno una volta: il dado da gioco.

Naturalmente ci riferiamo a un dado "ineccepibile" da un punto di vista delle possibilità che offre. In accordo alla teoria della probabilità, infatti, ogni faccia del cubo dovrebbe avere le stesse probilità di uscita: ciò significa che un dado è considerato "ideale" se, in seguito a 6000 lanci, ciascuna delle sei facce è comparsa circa 1000 volte. Il motivo di un numero di lanci talmente elevato è dovuto al fatto che la teoria della probabilità è valida solo se il numero delle possibilità (sei, nel nostro caso) sia di gran lunga inferiore a quello degli eventi (seimila, nell'esempio riporato).

Sarebbe sciocco, infatti, attendersi la comparsa di ciascuna delle sei facce effettuando altrettanti lanci! Purtroppo la gente non applica il ragionamento appena decritto, e la conseguenza di tale trascuratezza porta alla sciagurata teoria dei "ritardi" in base alla quale, ad esempio, se un certo numero del Lotto non viene estratto da un centinaio di settimane, dovrebbe avere una maggior probabilità di estrazione rispetto agli altri 89 numeri.
Un ritardo, infatti, a causa dell'elavato numero delle pedine (90), incomincia a diventare consistente dopo un migliaio di "mancate" uscite e non dopo qualche centinaio di settimane di assenza.

Il caso e il computer

Abbiamo accennato a due giochi di certo diversi, tenendo conto delle possibilità. Ma oltre al dado (6 facce), alla Tombola (90 pedine) vi sono moltissimi altri giochi (e non) in cui le possibilità cambiano in modo notevole. Le sole carte da gioco fanno venire in mente una gran varietà di casi per ciò che riguarda il numero di carte: 40 (napoletane), 52 (francesi), senza contare le "varietà" tra cui Canasta (due mazzi di carte), Poker (32 carte oppure 52) ed altri giochi in cui il numero varia considerevolmente.
Anche il Totocalcio, il Totip, l'Enalotto hanno la loro parte con tre possibilità (1, X, 2), per non parlare della "codificazione" di vari tornei che vedono impegnati giocatori e squadre variamente assortiti.
Come è possibile, dunque, utilizzare il computer per simulare un qualsiasi evento? Come, inoltre, progettare un sistema software in modo da accontentare chi si accinge a studiare fenomeni legati a possibilità non legate a giochi o eventi già noti?
Nella stesura del cosiddetto software di base, i progettisti dei calcolatori hanno provveduto ad inserire, nei linguaggi di programmazione (tra cui il Basic) uno "strumento" software in grado di soddisfare, con opportune manipolazioni, ogni esigenza.

Qualunque linguaggio, insomma, è in grado di generare un numero casuale il cui valore è compreso tra zero ed uno. Ve ne potete render conto facendo girare il seguente programma di una sola riga:

```
100 PRINT RND(0):GOTO 100
```

se la visualizzazione vi sembra che scorra troppo in fretta, potete premere il tasto CTRL (se possedete un Vic-20 o un C/64) oppure il tastino Commodore (primo tasto in basso a sinistra) nel caso possediate un C/16 oppure un Plus/4; se, poi, usate il C/128 in modo 128 potete premere il tasto No Scroll per bloccarla: premendolo nuovamente la visualizzazione riprenderà. Con il tasto Run/Stop, come dovreste già sapere, si interrompe il programma.

Ricordiamo che i numeri che compaiono sono tutti decimali perchè sono preceduti dal carattere punto (.). In effetti siamo abituati a scrivere 0,56... ma, per questioni di praticità, molti computer (e calcolatrici) preferiscono risparmiare la visualizzazione dello zero per consentire la visione di un maggior numero di cifre significative dopo la virgola (che però è un punto!..)

Cerchiamo, ora, di capire due cose: che significa RND e perchè la generazione di un numero compreso tra zero ed uno può andar bene in qualsiasi caso.

La funzione RND

Il nome RND rappresenta l'abbreviazione di Random che, in Inglese, vuol dire "a casaccio". La strana forma sintattica scelta per la funzione apparirà più chiara in seguito, quando ci accorgeremo che, in effetti, è possibile inserire, tra parentesi, non soltanto il numero zero ma anche altri valori tra cui, in particolare, uno (1) e valori negativi, come meno uno (-1).

Per ora ci limiteremo ad utilizzare la funzione con l'argomento nullo precisando che, in effetti, il valore che ne risulta non è realmente casuale, ma pseudo-casuale, dal momento che la sua elaborazione non è completamente slegata da even-

ti prevedibili (caratteristica peculiare del "vero" caso fortuito).

Il computer, infatti, assume, come base dei calcoli, l'orologio interno, la cui continua modifica (al ritmo di un aggiornamento ogni sessantesimo di secondo) è una garanzia sufficiente per una dignitosa imprevedibilità.

Approfondire il discorso sui numeri casuali e pseudo-casuali sarebbe fuori luogo, oltre che complesso, e ci limiteremo, nella presente trattazione, ad esaminare ed utilizzare la comoda funzione, presente, del resto, in qualsiasi linguaggio Basic utilizzato sui moderni calcolatori.

Se riprendiamo alcune regole di matematica, troveremo subito semplici applicazioni che consentiranno di "evadere" dal ristretto intervallo numerico prima visto (zero, uno).

Modifichiamo il semplice programma di prima come segue:

```
100 A=RND(0)
110 PRINT A*100
120 GOTO 100
```

Ai principianti (che farebbero bene, però, a prender nota della "scheda" presente in altra parte di questo inserto) raccomanderemo soltanto di non confondere il carattere alfabetico "O" con il numero zero "0", di premere il tasto Return alla fine della digitazione di ogni riga Basic e di far partire il programma digitando RUN e, subito dopo, il tasto return. Ricorderemo, inoltre, che per "variabile" si intende un gruppo di lettere alfabetiche (come "A", in riga 100 e 110) che vengono utilizzate come "contenitori" di numeri, risultati, elaborazioni, eccetera.

Facendo girare il programma, dicevamo, noteremo che i numeri visualizzati saranno ancora in forma decimale (con una parte numerica, cioè, presente dopo la virgola) ma compresi tra zero e 100. Con la riga Basic 110, infatti, non abbiamo fatto altro che moltiplicare per 100 il numero casuale generato dalla riga 100 e questo fatto, come è noto, corrisponde a "spostare" la virgola di due posti verso destra.

Volendo eliminare la parte decimale, possiamo ricorrere ad una comoda funzione che, appunto, restituisce la parte intera (INT) di un numero decimale.

Modificate il programma di prima come segue:

```
100 A=RND(0)
110 PRINT INT(A*100)
120 GOTO 100
```

Facendo girare il programma noteremo la visualizzazione continua di valori interi, compresi tra 0 e 100. In effetti il valore 100 non compare mai perchè anche se il valore RND valesse 0.99999999, questo, moltiplicato per 100, fornirebbe la cifra 99. Si potrebbe tentare di modificare 100 con 101 (riga 110) ma potrebbe capitare il caso in cui il valore determinato da RND sia 0.999999999 che porterebbe al valore intero di 101.

Nasce, così, il problema di individuare l'esatto intervallo di azione di RND, che può essere affrontato facendo girare, ad esempio, il seguente programma:

```
100 TI$="000000":Y=1
110 A=RND(0)
120 IF A>X THEN X=A: PRINT X
125 IF A<Y THEN Y=A: PRINT Y
130 GOTO 110
```

La prima riga (100) azzera l'orologio interno (ce ne potremo servire poi per conteggiare il tempo trascorso nella ricerca del valore massimo) ed assegna alla variabile Y il valore unitario; la seconda riga (110) genera un numero casuale. La riga 120 "associa" alla variabile X il valore elaborato di A solo se questo risulta maggiore di quello precedentemente memorizzato. La riga 125, al contrario, memorizza il valore di A solo se questo risulta più piccolo di quello contenuto in Y.

Non appena si fa girare il programma, alcuni numeri (tra cui il "minimo" zero) vengono subito visualizzati e in seguito sembra che non accada nulla. Il programma, però, sta girando (ad una velocità, tra l'altro, ragguardevole) ma i numeri casuali, pur fornendo almeno una volta il valore nullo, non sembrano raggiungere il valore 0.999999999. Dopo aver fatto girare il programma per oltre un'ora, abbiamo constatato che il valore più grande elaborato era:

```
0.996097267
```

E' probabile che, facendo girare il programma per un'intera

giornata, si pervenga a risultati diversi e tali, comunque, da mettere in difficoltà il programma che visualizzava numeri interi compresi tra 0 e 100.

Per ciòche ci riguarda, tuttavia, possiamo affermare che un C/64, sottoposto alla funzione RND, elabora un numero compreso tra zero e 0.996097267.

Abbiamo visto come sia semplice generare un numero casuale compreso tra 0 e 100. Ricorrendo ad analoghi ragionamenti, sarà possibile generare altri numeri appartenenti all'intervallo che più ci aggrada. A parte il caso banale dei valori compresi tra zero ed una potenza di dieci (10, 100, 1000, ...1000000 eccetera) vediamo di risolvere alcuni casi particolari:

Numeri casuali multipli

Ricordiamo brevemente che un numero si dice multiplo di un altro se, diviso per quest'ultimo, fornisce un risultato intero (privo, cioè di parte decimale). Esempi:

1768/4=442 (1768 è un multiplo di 4)
161/7=23 (161 è un multiplo di 7)
871/67=13 (871 è un multiplo di 67)

1768/3=589.333 (1768 non è un multiplo di 3)
161/5=32.2 (161 non è un multiplo di 5)
Eccetera

Se volete ottenere numeri casuali non maggiori di un certo valore e, contemporaneamente, multipli di un valore assegnato, è necessario attribuire tale valore ad una variabile di comodo (B, nell'esempio che segue) e moltiplicarla per il numero casuale moltiplicato, a sua volta, per il valore massimo desiderato. Digitate il seguente programma esplicativo:

```
100 A=RND(0)
110 B=7:REM MODULO
115 MA=100:REM MASSIMO
120 A=INT(A*MA/B)*B
130 PRINT A:GOTO 100
```

Questo programma visualizza di continuo un numero casuale intero compreso tra zero e 100 con la caratteristica di essere anche divisibile per 7.

Se modificate la riga 110 (assegnando a B un altro valore, come B=5) e, magari, anche la riga 115 (MA=1343), dopo il solito RUN potrete constatare la continua stampa di numeri interi, multipli di 5, che non superano mai il valore impostato 1343.

Numeri casuali anche negativi

In questo caso vogliamo scrivere un programma che prenda in considerazione anche i numeri negativi. Abbiamo visto, però, che vengon sempre generati numeri positivi e che il valore più piccolo è zero.

Non dobbiamo dimenticare, però, che la differenza tra un numero positivo ed uno negativo è rappresentata soltanto dal segno. Inoltre, nel linguaggio Basic, è possibile un'espressione del genere:

```
A=-A
```

che, apparentemente contraddittoria, serve per cambiare il segno ad un qualsiasi valore numerico: Esempio:

```
A=100:PRINT A:PRINT -A
```

Da questa esperienza deduciamo che la sintassi: A=-A significa esattamente: "Alla variabile A assegna ADESSO il valore che PRIMA possedeva la stessa variabile A, cambiato di segno".

Tornando al problema della generazione di numeri casuali positivi o negativi, provate a digitare il seguente listato:

```
100 A=RND(0)
105 B=RND(0)
110 IF B>0.5 THEN A=-A
115 PRINT A
120 GOTO 100
```

Come si può notare, si generano due numeri casuali; il primo (A) sarà il valore da visualizzare; il secondo, invece, (B) servirà soltanto per decidere il segno algebrico del primo. Il valore casuale di B, infatti, può esser compreso tra zero ed uno: possiamo ragionevolmente dedurre che la probabilità che sia generato un numero compreso tra zero e 0,5 sia all'incirca eguale a quella secondo cui sia compreso tra 0.5 e 1.

Con la riga 110 siamo "sicuri" che il 50% delle volte verrà generato un numero positivo e, di conseguenza, il 50% restante verrà generato un valore negativo. Guardando attentamente lo schermo, mentre il programma gira, è possibile constatare una presenza di segni negativi e positivi in egual misura.

...E quasi senza accorgercene abbiamo gettato le basi per una prima truffa! Provate a modificare la riga 110 come segue:

```
110 IF B>.7 THEN A=-A
```

Ora abbiamo costretto il programma a generare un segno negativo solo se RND(B) è maggiore di 0.7, vale a dire nel 30% dei casi. Variando a volontà la riga 110 potete fare in modo di osservare molto raramente un segno positivo (o negativo) alterando semplicemente il valore del confronto presente in riga 110.

Se si utilizza la tecnica descritta per far apparire più (meno) frequentemente una certa carta da gioco oppure la faccia di un dado, ecco che i giochi elettronici si prestano per le truffe più disparate: nessun oggetto può esser manipolato meglio delle macchinette mangiasoldi (slot machine, video poker, eccetera).

Tornando, però, ad argomenti meno... truffaldini, siamo sicuri che il lettore ha compreso perfettamente in che modo "estrarre" da un calcolatore un qualsiasi numero casuale, nella percentuale desiderata, nonostante il limite dell'intervallo compreso tra zero ed uno.

Numeri maggiori di un certo valore

Anche questo caso, sul quale ci intratteniamo molto brevemente, è basato su un banale ragionamento matematico: un

numero A è maggiore di un altro B se, sottraendo algebricamente B da A si ottiene un valore maggiore di zero.

Esempi:

```
100-25=75: 100>25
100-(-7)=107: 100>-7
-10-(-31)=21: -10>-31
```

Non dobbiamo trascurare, infatti, che in algebra un doppio segno negativo diventa positivo. Volendo, ad esempio, generare numeri casuali compresi tra 34 e 150, il listato che segue potrà svolgerne il compito:

```
100 A=RND(0)
110 B=34:REM MINIMO
120 C=150:REM MASSIMO
130 D=B+INT(A*(C-B+1))
140 IF D=B OR D=C THEN PRINT CHR$(18);
150 PRINT D;CHR$(146);:GOTO 100
```

Si noti la somma di "1" in riga 130, dovuta alla necessità di considerare anche il valore massimo (150, nel caso ipotizzato).

Approfitteremo del listato, comunque, per sottolineare la presenza del carattere di punto e virgola (;) nelle righe 140 e 150 che costringe il computer a visualizzare i numeri l'uno di seguito all'altro. Omettendo il carattere di punto e virgola, i valori casuali verrebbero stampati l'uno sotto l'altro, come nei listati finora visti.

Si noti, inoltre, la presenza dei due caratteri speciali:

CHR$(18) ha il compito di invertire i colori di ciò che verrà stampato in seguito.
CHR$(146) ha il compito di ripristinare i colori normali.

La presenza dei due caratteri speciali consentirà di individuare rapidamente su video, grazie al cambio del colore, l'eventuale "estrazione" sia del valore minimo (34) sia di quello massimo (150).

I listati da digitare

Siamo ora in grado di comprendere con facilità le varie istruzioni dei programmi che fanno parte del presente inserto. Li esamineremo, comunque, uno per uno mettendo in risalto le loro peculiarità sia per ciò che riguarda il compito che svolgono sia, soprattutto, per mettere in luce particolari tecniche di programmazione che il lettore potrà utilizzare in propri listati.

Ricordiamo ai lettori principianti di digitare i programmi così come sono pubblicati, riga per riga, e di premere il tasto Return alla fine di ogni riga, anche se sembra che non ve ne sia bisogno: le righe Basic vengono accettate dal computer soltanto se si preme quel tasto che, non a caso, è il più grande che si trova sulla tastiera.

Se, digitato il comando di partenza (RUN) compaiono segnalazioni di errore oppure il programma non gira come descritto nel paragrafo che lo commenta, tenendo premuto il tasto Run/Stop, premete il tasto Restore. Subito dopo digitate LIST e verificate che ciò che avete trascritto sia esattamente identico a ciò che compare sulla rivista. Se lo desiderate potete evitare di trascrivere tutto ciò che è presente sul rigo dopo il comando REM, in quelle righe Basic che, ovviamente, lo contengono.

1- Indovina un numero (a)

E' indubbiamente il programma più banale che possa capitare, anche ad un principiante. Ne approfitteremo, però, per commentarlo adeguatamente in modo da permettere ai lettori di comprenderlo nei minimi dettagli.

Nella riga 250 si assegnano a variabili stringa alcuni caratteri "speciali" tipici dei computer Commodore. Ricordando che con il simbolo:

⟨R⟩

si intende: "Premere il tasto Return", se digitate:

PRINT CHR$(65) ⟨R⟩

verrà visualizzato il carattere alfabetico "A". Ripetendo la prova vi accorgerete che ad ognuno dei 255 valori possibili (racchiusi tra parentesi) corrisponderà un determinato carattere (66=B, 67=C, eccetera).

Alcuni valori, però, non hanno alcun significato (0, 1, 12 ed altri). Altri ne hanno uno particolare: 144 cambia in nero il colore del cursore lampeggiante; 19 lo sposta in alto a sinistra sul video, 30 lo colora in verde, eccetera.

In particolare, con il valore 147 si cancella lo schermo e ne consegue che assegnare alla variabile stringa C1$ il carattere "speciale" 147, significa attribuire la capacità di eliminare qualsiasi carattere presente sul video tutte le volte che viene eseguito il comando PRINT C1$.

Analogamente, 17 è il codice per spostare il cursore in basso di un rigo e 18, invece, permetterà di invertire i colori dei caratteri che, in seguito, verranno stampati. Esempi:

```
PRINT CHR$(18)"PROVA" <R>
PRINT CHR$(147)"CANCELLO" <R>
PRINT CHR$(147)CHR$(18)"PIPPO" <R>
```

Allo stesso modo, nella riga 300, vengono assegnate ad altre quattro stringhe altrettanti messaggi, da far apparire in seguito a seconda dei casi verificati.

Con la riga 350, che cancella lo schermo e fa apparire un messaggio, inizia il gioco vero e proprio.

La riga 400 chiede l'esponente del numero dieci in modo da stabilire il "campo" in cui generare un numero casuale e, se esagerate oppure sbagialte a digitare, ripete la domanda. Non dimenticate di premere il tasto Return dopo aver battuto i valori numerici che il programma vi chiederà.

La riga 550 genera il numero casuale secondo le modalità viste in precedenza, mentre le righe successive rappresentano una sorta di "filtro" per la vostra risposta:

La riga 650, infatti, fa apprire il messaggio PI$ (cioè "Troppo piccolo") nel caso in cui il valore da voi digitato sia minore da quello scelto dal computer.

La riga 700 fa apparire il messaggio contenuto in GR$ nel caso contrario.

Solo se avete indovinato apparirà la stringa ES$ che conferma la vittoria.

```
100 REM INDOVINA UN NUMERO
150 REM PRIMA VERSIONE
200 :
250 C1$=CHR$(147):C2$=CHR$(17):C3$=CHR
    $(18):REM DEFINIZIONE STRINGHE
300 PI$="TROPPO PICCOLO":GR$="TROPPO G
    RANDE":ES$="ESATTO!":REM DEF.MESSA
    GGI
350 PRINTC1$C3$"INDOVINA UN NUMERO"C2$
400 INPUT "POTENZA DI 10 MAX (1/6)";PD
    :IF PD<1 OR PD>6 THEN 350:REM POTE
    NZA DI 10
450 PRINTC2$"O.K.: DEVI INDOVINARE UN
    NUMERO":REM DOMANDA
500 PRINTC2$"COMPRESO TRA 0 E"10↑PD
550 X=INT(RND(0)*10↑PD):REM GENERAZION
    E NUMERO CASUALE COMPRESO NELL'INT
    ERVALLO
600 PRINTC2$;:INPUT "DIGITA VALORE";NX
650 IF NX<X THEN PRINTPI$:GOTO 600
700 IF NX>X THEN PRINTGR$:GOTO 600
750 PRINTES$:REM INDOVINATO!
800 END
```

2- Indovina un numero (b)

Giocando con il programma precedente, e imponendo un esponente elevato al numero 10, ci accorgiamo di impiegare molto tempo per portare a termine il gioco. Questa nuova versione ha il compito di far capire al lettore in che modo è possibile intervenire su un programma per renderlo più interessante oppure più completo.

Trascurando di commentare le righe con caratteristiche simili a quelle viste nel listato precedente, faremo notare la presenza, stavolta, di più messaggi (righe 150 - 170) da visualizzare a seconda del valore digitato.

Si noti, inoltre, la comparsa della variablie Y (riga 240) che ha il compito di contare i tentativi effettuati per indovinare il numero.

Il lettore potrà studiare attentamente le righe 270-310 che, filtrando opportunamente la risposta del giocatore, visualizzano uno dei messaggi disponibili. Se, ad esempio, il valore digitato è maggiore di quello stabilito dal computer, non comparirà semplicemente, come nel programma di prima, il generico messaggio "Troppo grande", ma, a seconda della... lontananza dalla verità, comparirà "E' grandissimo", "E' troppo grande", "E' troppo grande ma ci sei vicinissimo!".

Una volta indovinato il numero compare anche il numero di tentativi compiuti.

```
100 REM INDOVINA UN NUMERO
110 REM SECONDA VERSIONE
120 :
130 C1$=CHR$(147):C2$=CHR$(17):C3$=CHR
    $(18):REM DEFINIZIONE CARATTERI SP
    ECIALI
140 PI$="TROPPO PICCOLO":GR$="TROPPO G
    RANDE":ES$="ESATTO!":REM DEFINIZ.M
    ASSAGGI
150 MI$=C3$+"E' PICCOLISSIMO":MA$=C3$+
    "E' GRANDISSIMO":REM DEF.MESSAGGI
160 V1$=C3$+"E' TROPPO PICCOLO, MA CI
    SEI VICINISSIMO!"
170 V2$=C3$+"E' TROPPO GRANDE, MA CI S
    EI VICINISSIMO!"
180 PRINTC1$C3$"INDOVINA UN NUMERO"C2$
190 INPUT "POTENZA DI 10 MAX (1/6)";PD
    :IF PD<1 OR PD>6 THEN 180:REM POTE
    NZA DI 10
200 PRINTC2$"O.K.: DEVI INDOVINARE UN
    NUMERO"
210 PRINTC2$"COMPRESO TRA 0 E"10↑PD
220 X=INT(RND(0)*10↑PD):IF X=0 THEN 22
    0:REM GENERAZIONE NUMERO CASUALE
```

```
230 REM Y=CONTATORE DI TENTATIVI EFFET
    TUATI
240 Y=Y+1:PRINTC2$;:INPUT "DIGITA VALO
    RE";NX:REM RICHIESTA NUMERO
250 IF X=NX THEN PRINTES$:PRINT"HAI IN
    DOVINATO DOPO"Y"TENTATIVI":END
260 IF NX>X*.9 AND NX<X THEN PRINTV1$:
    GOTO 240:REM DETERMINAZIONE ORDINE
     DI GRANDEZZA
270 IF NX<X*1.1 AND NX>X THEN PRINTV2$
    :GOTO 240
280 IF NX<X*.5 THEN PRINTMI$:GOTO 240
290 IF NX<X THEN PRINTPI$:GOTO 240
300 IF NX>X*1.5 THEN PRINTMA$:GOTO 240
310 IF NX>X THEN PRINTGR$:GOTO 240
320 END
```

3- Affidabilità del computer

Nella prima parte dell'inserto abbiamo visto che un sistema casuale è considerato tale solo se tutti i numeri possibili hanno la stessa probabilità di estrazione.

Il nostro computer possiede tali prerogative oppure qualche valore casuale viene generato più facilmente (difficilmente) di un altro?

Per stabilirlo affideremo allo stesso computer il compito di calcolare la sua attendibilità.

Dopo aver digitato (e verificato) il programma, battete, come al solito, RUN ⟨R⟩. Il calcolatore chiederà il numero di possibilità da esaminare.

Ciò significa che dovete digitare il numero di casi possibili: 6 per verificare l'attendibilità nel caso di giochi basati sul lancio dei dadi; 90 per la tombola; 3 per la schedina Totocalcio (1, 2, X); 40 per giochi con carte napoletane e così via.

Alla successiva domanda (Quante estrazioni?) dovete battere il numero di volte che il computer deve generare un numero casuale compreso nell'intervallo stabilito precedentemente.

Naturalmente tale valore deve essere in accordo con le leggi della probabilità, vale a dire almeno cento volte il numero dei casi possibili: 600 per i dadi, 9000 per la tombola, eccetera.

Dopo aver battuto il tasto Return, il computer passerà a generare i numeri casuali e visualizzerà di continuo il numero di tentativi. Al termine comparirà l'elenco delle estrazioni effettuate, con tanto di percentuali.

Si ricorda che la variabile X% non ha nulla a che fare con le percentuali, di cui possiede solo il simbolo. Questo modo un po' strano di indicare una variabile, serve a definire le variabili intere, vale a dire particolari valori numerici privi di parte decimale.

```
100 REM DETERMINAZIONE AFFIDABILITA' D
    ELLA CASUALITA' DI UN COMPUTER
110 :
120 PRINTCHR$(147):INPUT "N. POSSIBILI
    TA'";NP:IF NP<3 OR NP>100 THEN 120
130 DIM X%(NP):PRINTCHR$(147)
140 INPUT "QUANTE ESTRAZIONI";NE:IF NE
    <NP THEN 120
150 FOR I=1 TO NE
160 X=INT(RND(0)*(NP+1)):IF X=0 THEN 1
    60
170 X%(X)=X%(X)+1:PRINTCHR$(19)I:NEXT
180 PRINTCHR$(147)"N.     ESTRAZIONI
     PERCENTUALE":PRINT
190 FOR I=1 TO NP:PRINTI,CHR$(18)X%(I)
    ,(X%(I)*100)/NE:NEXT
200 PRINT:PRINT"SU UN TOTALE DI"NE"EST
    RAZIONI"
```

4- Minitombola

Questo programma non è che una variante di quello precedente e consente di generare, uno per uno, e per una sola volta, tutti i numeri determinati rispondendo alla prima domanda: "Numeri da estrarre?".

Se, ad esempio, digitate 90 a tale richiesta, noterete che, in seguito, i valori verranno estratti, in modo casuale, l'uno dopo l'altro e per una volta soltanto, fino a che non verrà generato l'ultimo valore.

Si noterà che i primi valori verranno visualizzati con notevole rapidità. Un po' per volta, però, tale velocità diminuirà sempre più perchè il calcolatore è costretto a scartare tutti i valori che, pur appartenendo all'intervallo impostato, sono già stati estratti.

```
100 REM MINITOMBOLA
110 :
120 FOR I=1 TO 4:K$=K$+CHR$(32):NEXT:R
    EM CREA STRINGA FORMATA DA 4 SPAZI
130 FOR I=1 TO 4:W$=W$+CHR$(157):NEXT:
    REM  CREA STRINGA DI 4 CRSR LEFT
140 Q$=K$+W$:REM SOMMA LE DUE STRINGHE
150 PRINTCHR$(147):INPUT "N. DA ESTRAR
    RE";NE:IF NE<3 OR NE>1000 THEN 150
160 PRINTCHR$(147)"VUOI UNA PAUSA TRA
    UN'ESTRAZIONE E L'ALTRA? (S/N)"
170 GOSUB 270:Y=0:IF A$="S" THEN Y=1
180 DIM X%(NE):PRINTCHR$(147) TAB(5)"←
     NUMERO DI ESTRAZIONI EFFETTUATE"
190 FOR I=1 TO NE:REM ESEGUE ESTRAZION
    I
200 X=INT(RND(0)*(NE+1)):IF X=0 OR X%(
    X)=1 THEN 200
210 X%(X)=1:IF Y=1 THEN GOSUB 310:REM
    ESAMINA SE ESTRAZIONE E' GIA' AVVE
    NUTA
220 PRINTCHR$(19)I:NEXT:PRINTCHR$(147)
230 REM FORI=1TONE:PRINTI;CHR$(18)X%(I
    ):NEXT
240 END
250 :
260 REM ESAME TASTO PREMUTO
270 GET A$:IF A$="" THEN 270
```

```
280 RETURN
290 :
300 REM VISUALIZZAZIONE NUMERO ESTRATT
    O
310 PRINT"NUMERO ESTRATTO"Q$;X
320 PRINT:PRINT"PREMI UN TASTO":GOSUB
    270:RETURN
330 END
```

5- Massimo & Minimo

Questo brevissimo listato serve solo a verificare che, in effetti, è possibile generare un qualsiasi valore numerico compreso all'interno di un intervallo stabilito.

Il programma, grazie al GOTO140 di riga 160, non si ferma mai (a meno che premiate il tasto Run/Stop) e visualizza di continuo i valori generati appartenenti all'intervallo prescelto. La riga 150 non viene mai "attivata" dal momento che non dovrebbe mai capitare il caso di una estrazione che esca dai limiti imposti.

In reverse appariranno, se estratti, i valori che coincidono con il minimo oppure con il massimo impostato.

```
100 REM NUMERO RANDOM COMPRESO TRA UN
    MINIMO ED UN MASSIMO
110 :
120 INPUT "VALORE MINIMO";V1
130 INPUT "VALORE MASSIMO";V2
140 X=V1+INT(RND(0)*(V2-V1+1)):IF X=V1
     OR X=V2 THEN PRINTCHR$(18);
150 IF X<V1 OR X>V2 THEN PRINT"ERRORE"
    :END
160 PRINTX;CHR$(146);:GOTO 140
```

6- Lancio di due dadi

Questo programma presenta una novità rispetto ai precedenti: il menu di scelta. Molti programmi sono, infatti, in grado di

svolgere numerosi compiti e tocca all'utente indicare la sua scelta.

In questo caso vi sono due possibilità: il lancio di due dadi (premendo il tasto 1) oppure il lancio di un solo dado.

Nel primo caso, ad ogni pressione di un tasto, verrà simulato un lancio, assolutamente casuale.

Se, invece, avete premuto il tasto 2, verrà posta una domanda sul tipo di computer che state adoperando. Tale precisazione è indispensabile poichè è necessario ricorrere a comandi POKE che, se mal utilizzati, rischiano di bloccare il computer. In casi come questi, comunque, tenete presente che non è possibile danneggiare il vostro amato calcolatore: al massimo sarete costretti a spegnerlo e riaccenderlo, perdendo, putroppo, tutto ciò che avevate digitato fino a quel momento. Ecco perchè è una buona norma quella di registrare un programma, terminata la digitazione, PRIMA di battere RUN!

Con la seconda opzione, tutte lè volte che premete un tasto verrà simulato il lancio di un solo dado, ma sotto forma semigrafica, ponendo cioè in reverse alcuni dei nove asterischi che compongono la faccia di un dado.

Gli argomenti delle istruzioni Poke rappresentano le locazioni di schermo dei vari computer, vale a dire le celle di memoria specifiche del video.

```
100 REM LANCIO DI DUE DADI
110 REM PER C/16, PLUS/4, COMMODORE 64
120 :
130 REM MENU PRINCIPALE
140 PRINTCHR$(147)"SCEGLI:":PRINT
150 PRINT"1- LANCIO DI DUE DADI"
160 PRINT"2- LANCIO DI UN DADO"
170 GOSUB 480:IF A$="1" THEN 220
180 IF A$="2" THEN 330
190 GOTO 140:REM ESCLUSIONE ALTRI TAST
    I
200 :
210 REM PRIMA OPZIONE: LANCIO DI DUE D
    ADI
```

```
220 PRINTCHR$(18)"PREMI UN TASTO PER I
    L LANCIO DEI DADI":PRINT
230 GOSUB 480:REM ESAME TASTO PREMUTO
240 GOSUB 300:PRINT"PRIMO DADO: "X:REM
    VISUALIZZA PRIMA ESTRAZIONE
250 GOSUB 300:PRINT"SECONDO DADO: "X:RE
    M VISUALIZZA SECONDA ESTRAZIONE
260 PRINT:GOTO 220:REM RICHIEDE NUOVO
    LANCIO
270 :X=INT(1+RND(0)*6):RETURN:REM DA U
    N MINIMO=1 A UN MASSIMO=6
280 :
290 REM GENERAZIONE LANCIO CASUALE
300 X=INT(1+RND(0)*6):RETURN:REM DA UN
     MINIMO=1 A UN MASSIMO=6
310 :
320 REM SECONDA OPZIONE
330 PRINTCHR$(147)"1- C-16, PLUS/4":PR
    INT"2- COMMODORE 64":GOSUB 480
340 PS=3072:IF A$="2" THEN PS=1024
350 PRINTCHR$(147):PRINT:PRINT:PRINT:P
    RINT:GOSUB 520:GOSUB 300:PRINTX
360 REM 1024 PRIMA LOCAZIONE DI SCHERM
    O PER COMMODORE 64
370 REM 3072 PRIMA LOCAZIONE DI SCHERM
    O PER C-16 E PLUS/4
380 IF X=1 THEN POKE PS+41,170:REM CAS
    O DEL PUNTEGGIO=1
390 IF X=2 THEN POKE PS+40,170:POKE PS
    +42,170:REM CASO DEL PUNTEGGIO=2
400 IF X=3 THEN POKE PS,170:POKE PS+41
    ,170:POKE PS+82,170:REM CASO DEL P
    UNTEGGIO=3
410 IF X=4 THEN POKE PS,170:POKE PS+2,
    170:POKE PS+80,170:POKE PS+82,170
420 IF X=5 THEN POKE PS,170:POKE PS+2,
    170:POKE PS+80,170:POKE PS+82,170:
```

```
    POKE PS+41,170
430 IF X=6 THEN POKE PS,170:POKE PS+2,
    170:POKE PS+80,170
440 IF X=6 THEN POKE PS+82,170:POKE PS
    +40,170:POKE PS+42,170
450 GOSUB 480:GOTO 350:REM ESAME PRESS
    IONE TASTO E SUCCESSIVO NUOVO LANC
    IO
460 :
470 REM ESAME TASTO PREMUTO
480 GET A$:IF A$="" THEN 480
490 RETURN
500 :
510 REM DISEGNO DEI SEI ASTERISCHI MED
    IANTE POKE
520 FOR I=0 TO 2:FOR J=0 TO 2:POKE PS+
    I+J*40,42:NEXTJ,I:RETURN
530 END
```

7- Poker

Questo gioco, che consente due scelte, stampa su video tutte le carte da Poker e verifica, graficamente, la casualità di estrazione delle carte stesse.

Con la seconda opzione, invece, compaiono i nomi dei semi ed il valore di un gruppo di cinque carte alla volta. Anche in questo caso, ovviamente, il computer evita che compaiano due carte eguali per ciascuna estrazione.

```
100 REM ESTRAZIONE DA UN MAZZO DI 52 C
    ARTE
110 :
120 DIM CA%(4,13,2)
130 A$="123456789DJQK"
140 PRINT"[CLEAR]POKER":PRINT:REM SELE
    ZIONE TIPO DI COMPUTER ADOPERATO
150 PRINT"1- COMMODORE 64"
```

```
160 PRINT"2- C/16 PLUS/4"
170 SC=983:GOSUB 460:IF X$="2" THEN SC
    =3031
180 PRINT"[CLEAR]POKER":PRINT:REM SCEL
    TA FORMA DEL GIOCO
190 PRINT"1- PRIMA VERSIONE"
200 PRINT"2- SECON.VERSIONE"
210 GOSUB 460:IF X$="1" THEN 260
220 IF X$="2" THEN 340
230 GOTO 180
240 :
250 REM PRIMA VERSIONE
260 PRINTCHR$(147)A$:PRINTA$:PRINTA$:P
    RINTA$:PRINT
270 PRINT"AD OGNI PRESSIONE DI UN TAST
    O CORRISPONDE L'ESTRAZIONE DI UNA
    CARTA"
280 X=INT(1+RND(0)*13):Y=INT(1+RND(0)*
    4)
290 A=SC+Y*40+X:IF PEEK(A)>128 THEN 28
    0
300 GOSUB 460
310 POKE A,PEEK(A) OR 128:U=U+1:IF U<5
    2 THEN 280
320 END
330 REM SECONDA VERSIONE
340 PRINTCHR$(147)
350 CLR :DIM CA%(4,13)
360 SE$(1)="CUORI":SE$(2)="QUADRI":SE$
    (3)="FIORI":SE$(4)="PICCHE"
370 DATA ASSO,DUE,TRE,QUATTRO,CINQUE,S
    EI,SETTE,OTTO,NOVE,DIECI,JACK,DONN
    A,RE
380 DIM CA$(13):FOR I=1 TO 13:READ CA$
    (I):NEXT
390 FOR J=1 TO 5
400 X=1:Y=13:GOSUB 450:CA=A:REM ESTRAZ
```

```
    IONE CARTA
410 X=1:Y=4:GOSUB 450:SE=A:REM ESTRAZI
    ONE SEME
420 IF CA%(SE,CA)=1 THEN 400
430 CA%(SE,CA)=1:PRINTSE$(SE),CA$(CA):
    NEXT
440 PRINT:PRINT"PREMI UN TASTO":GOSUB
    460:GOTO 340
450 A=X+INT(RND(0)*Y):RETURN
460 GET X$:IF X$="" THEN 460
470 RETURN
480 END
```

8- Oroscopo Random

Questo listato serve soltanto a prendere in giro i vostri amici dal momento che ignora completamente le risposte date alle varie domande e visualizza una risposta assolutamente casuale.

In effetti non è completamente vero che ignora qualsiasi risposta. Come potete notare (riga 140) la variabile Y viene posta ad 1 se il nome della persona è "Paola". In seguito (riga 210) il programma salterà ad una riga ben precisa nel caso in cui Y sia eguale ad uno (e cioè se la persona che desidera l'oroscopo è proprio Paola). Ne consegue che il lettore, cambiando a piacere la riga 140 (oppure digitando altre righe analoghe) potrà far apparire il messaggio che desidera, magari... personalizzato per ciascun amico che verrà invitato alla tastiera!

```
100 REM OROSCOPO RANDOM
110 REM COMMODORE 64, C/16, PLUS/4
120 :
130 INPUT "DIGITA IL TUO NOME";A$:IF A
    $="" THEN 130
140 IF A$="PAOLA" THEN Y=1:REM NOME CH
    E INTERESSA...
```

```
150 INPUT "DIGITA IL TUO MESE DI NASCI
    TA (1/12)";A$:X=VAL(A$)
160 IF X<1 OR X>12 THEN 150:REM ESCLUD
    E MESI "IMPOSSIBILI"
170 INPUT "DIGITA IL TUO ANNO DI NASCI
    TA (1900/1987)";A$:X=VAL(A$)
180 IF X<1900 OR X>1987 THEN 170:REM A
    CCETTA SOLO ANNO TRA 1900 E 1987
190 INPUT "DIGITA IL TUO GIORNO DI NAS
    CITA";A$:X=INT(VAL(A$))
200 IF X<1 OR X>31 THEN 190:REM ACCETT
    A SOLO GIORNO TRA 1 E 31
210 IF Y=1 THEN 350:REM SALTA A ROUTIN
    E VOLUTA SOLO IN PRESENZA DI "PAOL
    A"
220 :
230 REM GENERA NUMERO RANDOM
240 X=INT(1+RND(0)*4):REM 4=NUMERO DI
    FRASI
250 IF X<2 THEN 310:REM SALTA A ROUTIN
    E SPECIFICATA DA ESTRAZIONE CASUAL
    E
260 IF X<3 THEN 320
270 IF X<4 THEN 330
280 GOTO 340
290 :
300 REM MESSAGGI
310 PRINT"ARRIVERA' UN MESSAGGIO DA PR
    ENDERE IN GRAN CONSIDERAZIONE":END

320 PRINT"PRESTA LA MASSIMA ATTENZIONE
     ALLE FALSE AMICIZIE ":END
330 PRINT"PREVEDO UNA SETTIMANA MERAVI
    GLIOSA PER GLI AFFARI":END
340 PRINT"SFRUTTA QUESTO PERIODO: SI P
    RESENTANO MERAVIGLIOSE OCCASIONI":
    END
```

```
350 PRINT"VICINO A TE C'E' UNA PERSONA
     CHE TI VUOL MOLTO BENE ";
360 PRINT"MA TU NON TE NE ACCORGI...":
    END
```

9- Dadi truccati

Questo programma rappresenta un metodo per alterare i risultati di una estrazione casuale.

Si noti, tra l'altro, che la riga 220 e la 230 impongono il valore 6 a ciascuno dei due dadi nel caso si prema il tasto "H" oppure "K".

Analogamente la pressione della barra spaziatrice, o di uno dei due tasti del cursore, stabilisce la modifica del minimo, del massimo o di entrami i limiti. In tal modo chi conosce il programma può, apportando tutte le modifiche che desidera, alterare a piacimento i risultati dell'estrazione.

```
100 REM LANCIO DI DUE DADI
110 REM TRUCCATI
120 :
130 PRINTCHR$(147)CHR$(18)" L'USO DI Q
    UESTO PROGRAMMA PER SCOPI "
140 PRINTCHR$(18)"ILLECITI E' PUNIBILE
     A NORMA DI LEGGE":PRINT:PRINT
150 :
160 PRINTCHR$(18)"PREMI UN TASTO PER I
    L LANCIO DEI DADI":PRINT
170 :
180 REM ESAME TASTO PREMUTO
190 GET A$:IF A$="" THEN 190
200 :
210 Z=1:Y=6
220 IF A$="H" AND U1=0 THEN Z=6:Y=6:U1
    =1:REM TASTO H DA USARE UNA SOLA V
    OLTA
```

```
230 IF A$="K" AND U2=0 THEN Z=6:Y=6:U2
    =1:REM TASTO K DA USARE UNA SOLA V
    OLTA
240 IF A$=CHR$(32) THEN Z=2:Y=5:REM BA
    RRA SPAZIATRICE
250 IF A$=CHR$(17) THEN Z=4:Y=6:REM CU
    RSORE IN BASSO
260 IF A$=CHR$(29) THEN Z=1:Y=3:REM CU
    RSORE A DESTRA
270 GOSUB 320:PRINT"PRIMO DADO: "X
280 GOSUB 320:PRINT"SECONDO DADO: "X
290 PRINT:GOTO 160
300 :
310 REM GENERA NUMETRO RANDOM COMPRESO
     TRA 1 E 6
320 X=INT(Z+RND(0)*(Y-Z+1)):RETURN
330 END
```

10- Sequenza costante

Esiste una particolare forma dell'istruzione RND che permette di generare sempre la stessa sequenza di numeri, anche se si spegne il computer, si riaccende e si ricarica il programma.

E' sufficiente "dichiarare" all'inizio un valore negativo tra le parentesi di RND (riga 200 e 270). In seguito, ad ogni RND(1) (vedi riga 210 e 290), verrà riproposta la stessa sequenza determinata dal particolare valore negativo dichiarato all'inizio.

In pratica è possibile generare un'infinità di valori successivi casuali, ma tutti predeterminabili e legati ad un particolare valore negativo da tenere a mente.

Il programma dimostra quanto detto attraverso la visualizzazione della stessa sequenza di valori. Si noti che, con la prima opzione, la possibilità di generare, con RND(0), un valore realmente casuale non è impedita (parte finale di riga 210).

La seconda opzione permette di gettare le basi di un vero e proprio programma-truffa basato sul gioco dei dadi. Sarà suf-

ficiente imparare a memoria la sequenza dei lanci simulati per stabilire quando puntare somme consistenti.

La generazione di sequenze ripetibili, però, non è resa possibile per consentire truffe più o meno serie. In molti programmi di statistica, ad esempio, tale possibilità è preziosa per verificare il comportamento di alcune procedure computerizzate.

```
100 REM RIPETIZIONE STESSI NUMERI
110 :
120 PRINTCHR$(147)"SCEGLI:":PRINT
130 PRINT"1- VERIFICA RIPETIZIONE NUME
    RI"
140 PRINT"2- ESTRAZIONE DADI TRUCCATI"
150 GOSUB 350:IF A$="1" THEN 190
160 IF A$="2" THEN 260
170 GOTO 120:REM ESCLUDE ALTRI TASTI
180 :
190 PRINTCHR$(147);:REM CANCELLA SCHER
    MO
200 A=RND(-1):REM GENERA SEME
210 FOR I=1 TO 20:PRINTRND(1):NEXTI:PR
    INTCHR$(18)RND(0):REM GENERA SEQUE
    NZA
220 PRINTCHR$(17)"PREMI UN TASTO"
230 GOSUB 350:PRINTCHR$(19);:GOTO 200:
    REM  CHR$(19)=TASTO HOME
240 :
250 REM ESTRAZIONE DADI TRUCCATI
260 PRINT:PRINT"LA SEQUENZA DELLE ESTR
    AZIONI SARA' SEMPRE LA SEGUENTE:"
270 A=RND(-1):REM GENERA SEME
280 REM GENERA 20 LANCI CASUALI DI UN
    DADO
290 PRINT:FOR I=1 TO 20:PRINT1+INT(RND
    (1)*6);:NEXTI:PRINT
300 PRINT:PRINT"SE NON CI CREDI, PREND
    I NOTA DEI VALORI"
```

```
310 PRINT"E FAI RIPARTIRE IL PROGRAMMA
    "
320 END
330 :
340 REM ESAME TASTO PREMUTO
350 GET A$:IF A$="" THEN 350
360 RETURN
370 END
```

11- Generatore di compiti in classe

Questo lungo programma, che consigliamo di studiare con attenzione, può esser utile sia a studenti che ad insegnanti: permette di generare in modo casuale un certo numero di domande le cui risposte presuppongono l'elaborazione di calcoli matematici. Naturalmente vengono memorizzate anche le risposte esatte per facilitare la correttezza delle operazioni effettuate.

Pur avendo un valore molto limitato da un punto di vista didattico, questo programma può essere usato con successo nel caso in cui un insegnante voglia verificare rapidamente il livello di preparazione raggiunto dalla classe su un gruppo determinato di argomenti.

```
100 REM GENERATORE DI COMPITI IN CLASS
    E
110 REM COMMODORE 64, C-16, PLUS/4
120 :
130 PRINTCHR$(147)"OUTPUT SU VIDEO O S
    U STAMPANTE (V/S)"
140 REM SCELTA PERIFERICA: DN=3:VIDEO.
      DN=4:STAMPANTE
150 GOSUB 390:DN=3:IF X$="S" THEN DN=4
160 :
170 REM RICHIESTA TIPO DI ESERCIZI DA
    GENERARE
180 PRINTCHR$(147)"1- DIVISIONI (S/N)"
```

```
    :GOSUB 390:IF X$="S" THEN NC(1)=1
190 PRINT"2- PESO SPEC.(S/N)":GOSUB 39
    0:IF X$="S" THEN NC(2)=1
200 PRINT"3- TERZO LATO(S/N)":GOSUB 39
    0:IF X$="S" THEN NC(3)=1
210 :
220 REM RICHIESTA NUMERO DI COMPITI DI
    VERSI
230 PRINTCHR$(147);:INPUT "QUANTI COMP
    ITI";NC
240 IF NC(1)=1 THEN DIM  DV(2,NC):REM
    DIMENSIONA LE MATRICI NECESSARIE
250 IF NC(2)=1 THEN DIM  PS(5,NC)
260 IF NC(3)=1 THEN DIM  TL(5,NC)
270 :
280 REM INIZIO VISUALIZZAZIONE
290 OPEN 1,DN,7:FOR I=1 TO NC
300 PRINT#1,"-------------------------
    --------------------"
310 PRINT#1,"COMPITO N."I:PRINT#1,"---
    -------"
320 PRINT#1:GOSUB 410:PRINT#1:NEXT
330 IF NC(1)=1 THEN PRINT#1:PRINT#1,"D
    IVISONI":FOR I=1 TO NC:PRINT#1,I;D
    V(2,I):NEXT
340 IF NC(2)=1 THEN PRINT#1:PRINT#1,"P
    ESO SPEC.":FOR I=1 TO NC:PRINT#1,I
    ;PS(5,I):NEXT
350 IF NC(3)=1 THEN PRINT#1:PRINT#1,"T
    ERZO LATO":FOR I=1 TO NC:PRINT#1,I
    ;TL(2,I):NEXT
360 PRINT#1:CLOSE 1:END
370 :
380 REM ESAME TASTO PREMUTO
390 GET X$:IF X$="" THEN 390
400 RETURN
410 IF NC(1)=1 THEN GOSUB 470:REM ESEC
```

```
    UZIONE ROUTINE N.1 (SE RICHIESTA)
420 IF NC(2)=1 THEN GOSUB 580:REM ESEC
    UZIONE ROUTINE N.2 (SE RICHIESTA).
     ECCETERA
430 IF NC(3)=1 THEN GOSUB 740
440 RETURN
450 :
460 REM DIVISIONI
470 PRINT#1,"ESEGUIRE LA DIVISIONE:";
480 A=300:X=100000:GOSUB 920:DV(0,I)=Y
    /100:PRINT#1,DV(0,I);:REM DIVIDEND
    O
490 A=0:X=10:GOSUB 920:DV(1,I)=INT((Y/
    7)*100)/100:REM DIVISORE
500 REM ESCLUSIONE CASI BANALI
510 DV(1,I)=INT((Y/7)*100)/100:IF DV(1
    ,I)=0 OR DV(1,I)=1 OR DV(1,I)=0.1
    THEN 490
520 PRINT#1,"/";
530 IF DV(1,I)<1 THEN PRINT#1,"0";
540 PRINT#1,DV(1,I)
550 DV(2,I)=DV(0,I)/DV(1,I):PRINT#1:RE
    TURN
560 :
570 REM PESO SPECIFICO
580 PRINT#1:PRINT#1,"UNA TAVOLA DI LEG
    NO MISURA":PRINT#1,"CM:";
590 REM 0,I=PRIMO LATO. 1,I=SECONDO LA
    TO. 2,I=TERZO LATO. 3,I=VOLUME
600 REM 4,I=PESO. 5,I=PESO SPECIFICO
610 A=10:X=20:GOSUB 920:PRINT#1,Y;:PS(
    0,I)=Y:PRINT#1,"  CM:";
620 A=20:X=60:GOSUB 920:PRINT#1,Y;:PS(
    1,I)=Y:PRINT#1,"  CM:";
630 A=1.5:X=5:GOSUB 920:PRINT#1,Y;:PS(
    2,I)=Y
640 PS(3,I)=PS(0,I)*PS(1,I)*PS(2,I)
```

```
650 PRINT#1,"  E PESA GRAMMI";
660 A=4:X=10:GOSUB 920
670 PS(4,I)=INT(PS(3,I)*Y/7)
680 IF PS(4,I)/PS(3,I)=1 OR PS(4,I)/PS
    (3,I)>1 THEN 660
690 PRINT#1,PS(4,I)
700 PRINT#1,"QUALE E' IL SUO PESO SPEC
    IFICO MISURATO IN KG/M3?":PRINT#1
710 PS(5,I)=1000*PS(4,I)/PS(3,I):RETUR
    N
720 :
730 REM TERZO LATO
740 PRINT#1,"UNA TAVOLA DI LEGNO MISUR
    A"
750 REM 0,I=PRIMO LATO.  1,I=SECONDO L
    ATO. 2,I=TERZO LATO
760 PRINT#1,"CM:";
770 A=10:X=20:GOSUB 920:PRINT#1,Y;:TL(
    0,I)=Y
780 PRINT#1,".  CM.:";
790 A=20:X=60:GOSUB 920:PRINT#1,Y;:TL(
    1,I)=Y
800 PRINT#1,"DETERMINARE LA LUNGHEZZA
    DEL TERZO LATO SAPENDO CHE"
810 A=1.5:X=5:GOSUB 920:TL(2,I)=Y
820 TL(3,I)=TL(0,I)*TL(1,I)*TL(2,I)
830 A=0:X=10:GOSUB 920:IF Y=0 THEN 830
840 TL(4,I)=INT(TL(3,I)*Y/7)
850 TL(5,I)=INT(1000*TL(4,I)/TL(3,I)):
    IF TL(5,I)>800 THEN 830
860 IF TL(5,I)<400 THEN 830
870 PRINT#1,"PESA G.:"TL(4,I);
880 PRINT#1,"ED IL PS VALE KG/M3";
890 PRINT#1,TL(5,I):RETURN
900 :
910 REM GENERAZIONE NUMERO CASUALE
920 Y=A+INT(RND(0)*X):RETURN
930 END
```

Scheda per i principianti

E' facile perdersi d'animo dopo i primi, inevitabili insuccessi con un computer.

Riportiamo qui di seguito il numero dei fascicoli di Commodore Computer Club in cui sono apparsi articoli di un certo interesse per chi si avvicina per la prima volta ad un calcolatore.

N.22: Come realizzare un menu
N.23: Microgioco da addestramento
N.24: Come realizzare uno spreadsheet
N.25: Variazioni demografiche
N.26: Spaghetti Basic
N.27: Tempo di saldi
N.27: La merce più economica
N.27: Impariamo a sbagliare
N.27: Utilizzo del Joystick
N.28: Il tuo primo programma
N.28: La tua prima volta con la grafica del C/64
N.29: 20 domande sul linguaggio macchina
N.29: Costruisci un labirinto
N.30: Tutti i caratteri del C/64
N.31: E non indurci in Syntax Error
N.32: Gli errori dovuti alla manipolazione delle stringhe
N.34: La programmazione modulare (a)
N.35: La programmazione modulare (b)
N.36: Intelligenti deduzioni artificiali

Per procurarsi i fascicoli arretrati leggere le modalità riportate in altra parte della rivista.